AD NUMBER

AD913118

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; 18 MAY 1971. Other requests shall be referred to Electronic Systems Division, L. G. Hanscom field, Bedford, MA 01730.

AUTHORITY

ESD ltr, 4 Dec 1973

Project Report

PA-229-2
(RSP)

Data Reduction Program Documentation
ALTOAK

(Effective: March 1971)

C. R. Berndtson
R. H. French
D. E. Nessman

19620

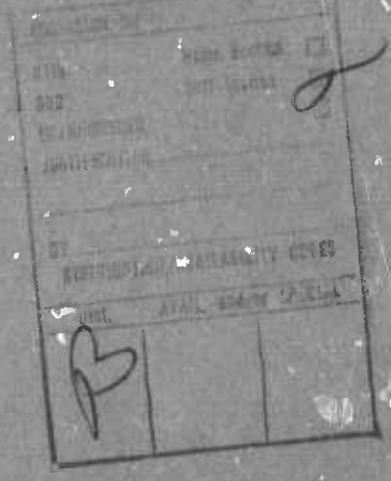
17 March 1971

Prepared for the Advanced Research Projects Agency,
the Department of the Army, and the Department of the Air Force
under Electronic Systems Division Contract F19628-70-C-0230 by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Lexington, Massachusetts



ERRATUM SHEET

for

PROJECT REPORT PA-229-2

Page 2:

The channel on the transcription tape labeled Az error is in reality traverse error. Therefore, please replace Paragraph 2 with the following:

The VHF LC RCS is computed for all requested range gates. The gate with the peak RCS is identified, and the RCS in the Az and El error channels for this gate are computed. The RCS differences

$$\Delta Tr \text{ (db)} = \text{Az error channel RCS} - \text{VHF LC RCS}$$
$$\Delta El \text{ (db)} = \text{El error channel RCS} - \text{VHF LC RCS}$$

are computed and used to index prestored tables of ΔTr (deg) vs ΔTr (db) and ΔEl (deg) vs ΔEl (db). The magnitude of the angle offsets [ΔTr (deg) and ΔEl (deg)] is then given. The azimuth offset is computed:

$$\Delta Az \text{ (deg)} = \Delta Tr \text{ (deg)}/\cos El \text{ (deg)}.$$

The label on the transcription tape, Az error channel RCS, has not been changed.

Use This Errata Sheet

17 May 1971

Publications Office
M.I.T. Lincoln Laboratory
P.O. Box 73
Lexington, Massachusetts 02173

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

6 DATA REDUCTION PROGRAM DOCUMENTATION ALTOAK (EFFECTIVE: MARCH 1971),

10 C. R. BERNDTSON,

[redacted]2

R. H. FRENCH
D. E. NESSMAN

Philco-Ford Corporation
Editors

12 64 p.

14

9 PROJECT REPORT, PA-229-2 (RSP)

11 17 MAR[redacted]71

15 F19628-70-C-0234,
ARPA Order-600

L. G. Hanscom Field
Bedford, Mass. 01730

207 650

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored by the Advanced Research Projects Agency of the Department of Defense (ARPA Order 600), the Department of the Army, and the Department of the Air Force under Air Force Contract F19628-70-C-0230.

FOREWORD

This is the second report in the Data Reduction Program Documentation series. It is dated according to the date of completion of the documentation. No implication is made that this program will not subsequently be modified, amended, or superseded: on the contrary, the history of radar data processing is one of continuous evolution of techniques, and it is unrealistic to assume that steady-state has been reached. The PA-229 series is being published for the convenience of interested parties, and Lincoln assumes no responsibility for the correctness of the information presented, nor for its currency.

The preparation of reports in this series is under the Editorship of Charles R. Berndtson of Lincoln, and of D. Nessman and R. French of Philco-Ford Corporation. Inquiries, suggestions, corrections, criticisms, and requests for additional copies should be directed to C. R. Berndtson.

The principal contributor to this report was A. J. Poirier (Philco-Ford). Due to the intricate, evolutionary manner in which the programs came into being, the editors regret that it is in general impossible to give due credit to all -- mathematicians or radar analysts or programmers -- who contributed to the definition and writing of the programs.

Alan A. Grometstein

CONTENTS

CONTENTS (cont'd)

ALTOAK

I. PURPOSE AND UTILIZATION

A. Source of data

ALTAIR[1]

B. Data Input

ALTAIR transcription tape

C. Description

ALTOAK is used to analyze data on targets which were not in angle track. For any object in range track by the ~~ARS sytem~~, the program computes for selected averaging intervals the off-axis angle coordinates (~~ΔAz~~, ΔEl), the total off-axis angle (~~θ~~), and the ~~RCS~~-corrections at VHF and UHF. The angle coordinates are obtained by comparing the VHF ~~LC~~ RCS with the RCS in the ~~Az~~ and ~~El~~ error channels. The corresponding phases are used to determine the sense of ~~ΔAz and ΔEl~~.

D. Output

1. A listing of all computed quantities.
2. Plots vs TAL of the uncorrected and corrected RCS at VHF LC (Optional: punched cards containing the RCS corrections).
3. Punched cards containing R, Az, and El, corrected for known errors, in a format suitable for input to NRTPOD.

II. DESCRIPTION

The following computations are performed each averaging interval.

The VHF LC RCS is computed for all requested range gates. The gate with the peak RCS is identified, and the RCS in the Az and El error channels for this gate are computed. The RCS differences

$$\Delta Az \text{ (db)} = \text{Az error channel RCS} - \text{VHF LC RCS}$$
$$\Delta El \text{ (db)} = \text{El error channel RCS} - \text{VHF RC RCS}$$

are computed and used to index prestored tables of ΔAz (deg) vs ΔAz (db) and ΔEl (deg) vs ΔEl (db). The magnitude of the angle offsets (ΔAz (deg) and ΔEl (deg)) is then given.

The pulse by pulse phase of the Az and El error channels are compared with the VHF LC phase to produce:

$$PHAZ = \frac{1}{N} \sum_{i=1}^{N} (\text{Az phase} - \text{VHF LC phase})$$

$$PHEL = \frac{1}{N} \sum_{i=1}^{N} (\text{El phase} - \text{VHF LC phase})$$

where N is the number of pulses in the averaging interval.

The sign of ΔAz (deg) is positive when cos PHAZ is positive and the sign of ΔEl (deg) is positive when cos PHEL is positive.

The total off-axis angle (θ) is found by:

$$\theta = \left[\Delta El^2 + \Delta Az^2 \cos^2 El \right]^{\frac{1}{2}}$$

VHF and UHF RCS corrections are determined from tables of the VHF and UHF beam shapes. Plots of corrected and uncorrected VHF LC RCS vs TAL are produced.

The following options are also available:

1. Punched cards containing TAL and the VHF and UHF RCS corrections.

2. Punched cards in a format acceptable to NRTPOD may be obtained containing R, Az, and El corrected for certain errors. R is corrected for bias, tropospheric refraction, and target position in sampling pattern. El is corrected for bias, tropospheric refraction, and ΔEl. Az is corrected for bias and ΔAz. Ionospheric refraction corrections are not made.

A number of input parameters and transcription tape parameters are checked for validity before processing.

The main program checks the following input parameters:

IPAT	=	1 or 2
TAVG	$\neq$	0
INTARG	$\neq$	0
NRG	$\neq$	0

Subroutine ALREAD[2] makes a number of other checks on transcription tape parameters. For some errors (missing format tables; end of file; target no., sampling pattern, or polarization not on tape) information is returned to the main program for decision to terminate.

III. OPERATION

A. Input

Start and stop times (GMT)

Averaging interval and skip time*

Target and sampling pattern numbers

Specified set of range gates

In addition, punched card output may be requested containing RCS corrections or observation data for input to NRTPOD. A sample ALTOAK input is shown in Appendix A.

CARD 1 (15A4)

(Col.)		
1-60	TITLE	60 character title for printout and plots

CARD 2 (2 (2I3, F7.3), 4X, 4I5, 2F10.3, 2I5)

(Col.)		
1-3	IH1 (I3)	Start time (GMT) in h, min, and s
4-6	IM1 (I3)	
7-13	ZSEC1 (F7.3)	
14-16	IH2 (I3)	Stop time (GMT) in h, min, and s
17-19	IM2 (I3)	
20-26	ZSEC2 (F7.3)	
31-35	NRG	Number of range gates (I5)
36-40	INTARG	Target no. (I5)

41-45	IPAT	Sampling pattern in which initial gate is located (I5)
46-50	ING*	Location within IPAT of initial gate (I5)
51-60	TAVG	Averaging interval in seconds (F10.3)
61-70	TSKIP	Skip time* in seconds (F10.3)
71-75	ICARD	1 = punch cards with VHF/UHF RCS corrections
		0 = no punch
76-80	INRT	1 = punch cards for NRTPOD
		0 = no punch
CARD 3	(3I5)	If INRT = 0 Card 3 must not be included
1-5	IYEAR	Last two digits of year
6-10	IMONTH	Month (1 to 12)
11-15	IDAY	Day of month

B. Output

LISTING

GMT

Az and El corrected for bias

VHF LC RCS for peak gate

Az and El error RCS for peak gate

ΔAz, ΔEl, and θ

VHF and UHF RCS corrections (db)

PHAZ and PHEL

CRLC (corrected VHF LC RCS)

RFRANG, AZCR, RFELV (corrected values of R, Az, and El for input to NRTPOD)

*Called ISG in program listing, and ISTGAT in ALREAD.

**Skip time is the time in seconds from the end of one averaging interval to the start of the next.

PLOTS

Peak gate RCS vs TAL

Peak gate RCS corrected for off-axis position vs TAL (Both standardized to 4 s/in for the abscissa and 20 db/in for the ordinate).

PUNCHED CARDS

RCS correction data:

TAL (F10.3)
ΔVHF RCS (db) (F10.3)
ΔUHF RCS (db) (F10.3)

Observation data (NRTPOD):

Radar identification (A3)
Year (3X, I2)
Month (I2)
Day (I2)
h (I2)
min (I2)
s (I2)
ms (I5)
Orbit no. (1X, I1)
Az (F8.3)
El (4X, F8.3)
R (4X, F12.4)

R, Az, and El are related to the middle pulse of an averaging interval.*

*They are determined in ALREAD as follows:

$$R = R_{t_0} + \dot{R}_{t_0} (t - t_0)$$

where R_{t_0}, $\dot{R}_{t_0}$ are R and $\dot{R}$ at the first pulse in the minor cycle

t_0 is the time of the first pulse in the minor cycle

t is the time of the middle pulse

Az and El values are available every 25 ms. The value closest to t in the major cycle containing the pulse is used.

The middle pulse is determined by the largest integer in $(N + 1)/2$, where N is the number of pulses in the averaging interval. Other output quantities are associated with the median time of the averaging interval, determined by $(T_{\text{last pulse}} - T_{\text{first pulse}})/2$. Sample ALTOAK outputs are shown in Appendix B.

IV. PROGRAM LIMITATIONS

A. General Remarks

When using ALTOAK, the following precautions should be observed.

The noise level should be determined theoretically or by examining a gate that does not contain a target. The signal in the VHF LC, Az, and El channels should be > 5 db above this noise level to obtain valid results. If the error channel signal is at or just above the noise level, the computed angle offset is an upper bound for the actual offset.

When correcting RCS for off-axis position, a rule of thumb frequently used is that the cross section correction must be ≤ 6 db. If the correction is > 6 db, the accuracy of the correction is questionable.

The user should be careful when using ALTOAK for trajectories that have high Az and El rates. Due to the method used in ALREAD for picking off the Az and El associated with a pulse, a lag of as much as 50 ms may result. This lag, which will vary in a saw tooth shape, probably will be small compared to the noise in the Az and El offset angles for trajectories with small angular rates. The RCS corrections and off-axis position data are not affected.

B. Limits of Parameters

Start Time	Must be on tape
Stop Time	Must be on tape
NRG	≤ 30 gates
TAVG	Must be larger than the PRI
TSKIP	Can not be negative
INTARG	Must be on tape within start and stop times
Length of Run	≤ 1200 averaging intervals

V. PROGRAMMING

A. OAKOS (see Appendices C and D.)

OAKOS is the control section of ALTOAK. OAKOS reads the input cards and calls all of the subroutines that process, plot, and print the data.

B. GLMP (see Appendices E and F.)

GLMP searches the array of gates selected to find the one with the largest RCS for each time interval. GLMP then computes and saves the corresponding PHEL, PHAZ, Az channel RCS, El channel RCS, and gate number. GLMP then inputs these values to SUBOAK along with GMT, R, Az, and lift-off time.

The call statement is GLMP (AVGAL, ISTGT, AVGAZ, AVGEL, IGAT, AVGTM, AVGRG, NRG, INDEX, ICARD, TLIFT).

INPUT

AVGAL	Alt*
AVGAZ	Az*
AVGEL	El*
AVGTM	Time* (GMT)
AVGRG	R*
ISTGT	Start gate for peak
NRG	Number of range gates
ICARD	Flag for punched cards
TLIFT	Lift-off time

OUTPUT

INDEX	Number of correction cards punched
IGAT	Peak gate number

*Midpoint in averaging interval.

C. TSPLIT (see Appendix G.)

GLMP calls TSPLIT. TSPLIT is used to convert time from total GMT to h, min, s, and decimal fractions of s.

The call statement is TSPLIT (AVGTM, IHM, TRUN).

INPUT

AVGTM	GMT total seconds

OUTPUT

IHM (1)	Hours
IHM (2)	Minutes
TRUN	Seconds and decimal fractions of seconds

D. SUBOAK (see Appendices H and J.)

GLMP calls SUBOAK. SUBOAK is the routine in which most of the computation is done. It consists mainly of pre-stored tables and equations to compute ΔAz, ΔEl, and $\Delta\theta$. SUBOAK computes correction factors for UHF and VHF RCS. SUBOAK also prints the data and punches the UHF/VHF correction cards.

The call statement is SUBOAK (IHR, IMIN, ZSEC, GMAX, AVGAL, IGAT, AVGTM, AZE, ELEX, PHAZ, PHEL, Az, El, INDEX, AVGRG, ICARD, TLIFT).

INPUT

IHR	Hours (GMT)
IMIN	Minutes (GMT)
ZSEC	Seconds (GMT)
GMAX	RCS for peak gate
AVGAL	Alt for midpoint of averaging interval
AVGTM	GMT for midpoint of averaging interval

AVGRG	R for midpoint of averaging interval
IGAT	Peak gate number
AZE	Az
ELEX	El
PHAZ	Az phase - VHF LC phase
PHEL	El phase - VHF LC phase
Az	Az channel RCS
El	El channel RCS
ICARD	Flag for punching correction cards
TLIFT	Lift-off time

OUTPUT

INDEX	Number of correction cards punched

STORED IN COMMON

TIM	TAL
TEM	Seconds and fractions of seconds (portion of GMT)
CRLC	Corrected VHF LC RCS
GTMAX	Uncorrected VHF LC RCS
RRANG	R corrected for tropospheric refraction
RELEV	El corrected for tropospheric refraction and off-axis position
AZCR	Az corrected for off-axis position

E. REFC (see Appendix K.)

SUBOAK calls REFC.

The tropospheric refraction correction subroutine, REFC, is based on tropospheric refraction tables in PPP-36.[3] A modified version of this subroutine is now in use.

The call statement is REFC (E, R, DEE, DRR)

E	Uncorrected El (must be between 0° and 90°)
R	Uncorrected R (ft)
DEE	El tropospheric correction
DRR	R tropospheric correction (ft)

The corrected values to be computed after exiting from the REFC routine are:

El	=	E - DEE
R (ft)	=	R - DRR

F. OAKPLT

OAKPLT is a plot routine which is called twice from OAKOS, once to plot the uncorrected VHF LC RCS vs TAL and once to plot the corrected VHF LC vs TAL. The user has no control over the size of the plot.

G. REW

REW is an entry to subroutine BREADS[4] used to rewind the tape.

H. ALREAD[2]

ALREAD is the Fortran driver for the assembler language tape reading routines. Appendix L presents an ALREAD program listing which has minor differences from the listing in Ref. 1. These differences occur because:

1. ALTOAK requires VHF LC, Az and El error channel RCS data. Only one channel per run is used by the other versions of ALREAD.

2. Only ALTOAK requires Az and El data.

The call statement is ALREAD (TSTART, TSTOP, TLIFT, INTARG, INPAT, NOPHA, NPTS, NFP, NEWPAS, NRG, ISTGAT).

INPUT

TSTART	Start time of processing (GMT total seconds)
TSTOP	End time of processing (GMT total seconds)
INTARG	Target number to be processed
INPAT*	Sampling pattern in which initial gate is located
NOPHA	2 (phase and RCS data wanted)
NRG	Number of range gates to be processed
ISTGAT**	Location within INPAT of initial gate

INPUT AND OUTPUT PARAMETERS

NPTS†	Output: number of pulses of data returned
	Input: must be initialized by calling program before each call to ALREAD

*Also called IPAT.

**Also called ING or ISG.

†Set to zero for first call. Set to number of saved points for subsequent calls.

NEWPAS*	Cycle and error pointer (see error returns and switch settings)

OUTPUT

TLIFT	Lift-off time (GMT total seconds)
NFP	Frequency code: 1 = VHF; 2 = UHF

STORED IN COMMON

TIMES	Pulse times (GMT total seconds)
XSPHA	RCS and phase for each pulse and gate
RANGKM	R
ALSAV**	Alt
IRGA	Range gate array associated with XSPHA
AZI	Az (rad)
ELE	El (rad)
IPOL	Data channels wanted
NPOL	Number of data channels used

J. Plotting System Subroutines

The subroutines are REREAD, STOIDV, and PLTND.

*Also called IAGAIN.

**Valid only for first pulse of minor cycle. It is repeated for subsequent pulses.

REFERENCES

1. "ALTAIR Data User's Manual", LM-97, Lincoln Laboratory, M.I.T. (to be published), UNCLASSIFIED.

2. "Data Reduction Program Documentation, ALREAD, (Effective: March 1971)", PA-229-3, Lincoln Laboratory, M.I.T. (17 March 1971), UNCLASSIFIED.

3. J. P. Penhune, "Refraction Corrections for the TRADEX Radar", PPP-36, Lincoln Laboratory, M.I.T. (21 April 1965), UNCLASSIFIED.

4. "Data Reduction Program Documentation, ALTAIR Tape Read Package, (Effective: April 1970)", PA-229-1, Lincoln Laboratory, M.I.T. (17 March 1971), UNCLASSIFIED.

COMMON SYMBOLS AND ABBREVIATIONS

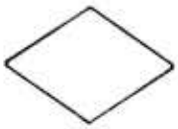
(The units given for certain quantities are the units commonly used for those quantities, unless otherwise noted.)

ADT	ALCOR Data Tape
Alt	Altitude (km)
APS	Average Pulse Shape
ARS	ALTAIR Recording System
Avg	Average, Averaging
Az	Azimuth (deg)
CADJ	Adjusted Calibration Constant (db)
C-band	ALCOR frequency, 5664 MHz (NB) and 5667 MHz (WB)
El	Elevation (deg)
EOF	End of File
GMT	Greenwich Mean Time
h	Hours
Hz	Hertz
in	Inches
LC	Left Circular Polarization
min	Minutes
NB	Narrow Band
NRTPOD	Non-real Time Precision Orbit Determination Program
POD	Project PRESS Operation and Data Summary Report
Phase	Presented in deg
PRF	Pulse Repetition Frequency (pps)
PRI	Pulse Repetition Interval (s)
pps	Pulses per second
pts	Points

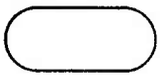
R	Range (km)
$\dot{R}$	Range Rate (km/s)
rad	Radians
RC	Right Circular Polarization
RCS	Radar Cross Section (dbsm)
s	Seconds
SD_w	Standard Deviation of Wake Velocity
T	Time
TAL	Time After Launch (s)
UHF	ALTAIR Frequency; 415 MHz
V	Velocity
V_d	Doppler Velocity
V_w	Mean Wake Velocity
VHF	ALTAIR Frequency; 155.5 MHz
WB	Wide Band
θ	Total Off-axis Angle (deg)
λ	Wavelength
*	Denotes Multiplication

FLOW DIAGRAM SYMBOLS

PROCESS, ANNOTATION



DECISION

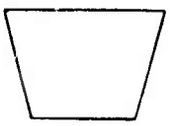


TERMINATOR

SUBROUTINE: where NAME is the entry call into the subroutine

CONNECTOR: where P specifies a page in the flow diagram, and L designates a statement number in the program listing or a reference point in the flow diagram

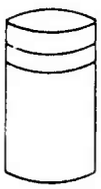
CONNECTOR: where X implies a continuation of the diagram to the next page



INPUT/OUTPUT OPERATION

MAGNETIC TAPE

PUNCHED CARD



DISK

APPENDIX A
ALTOAK INPUT

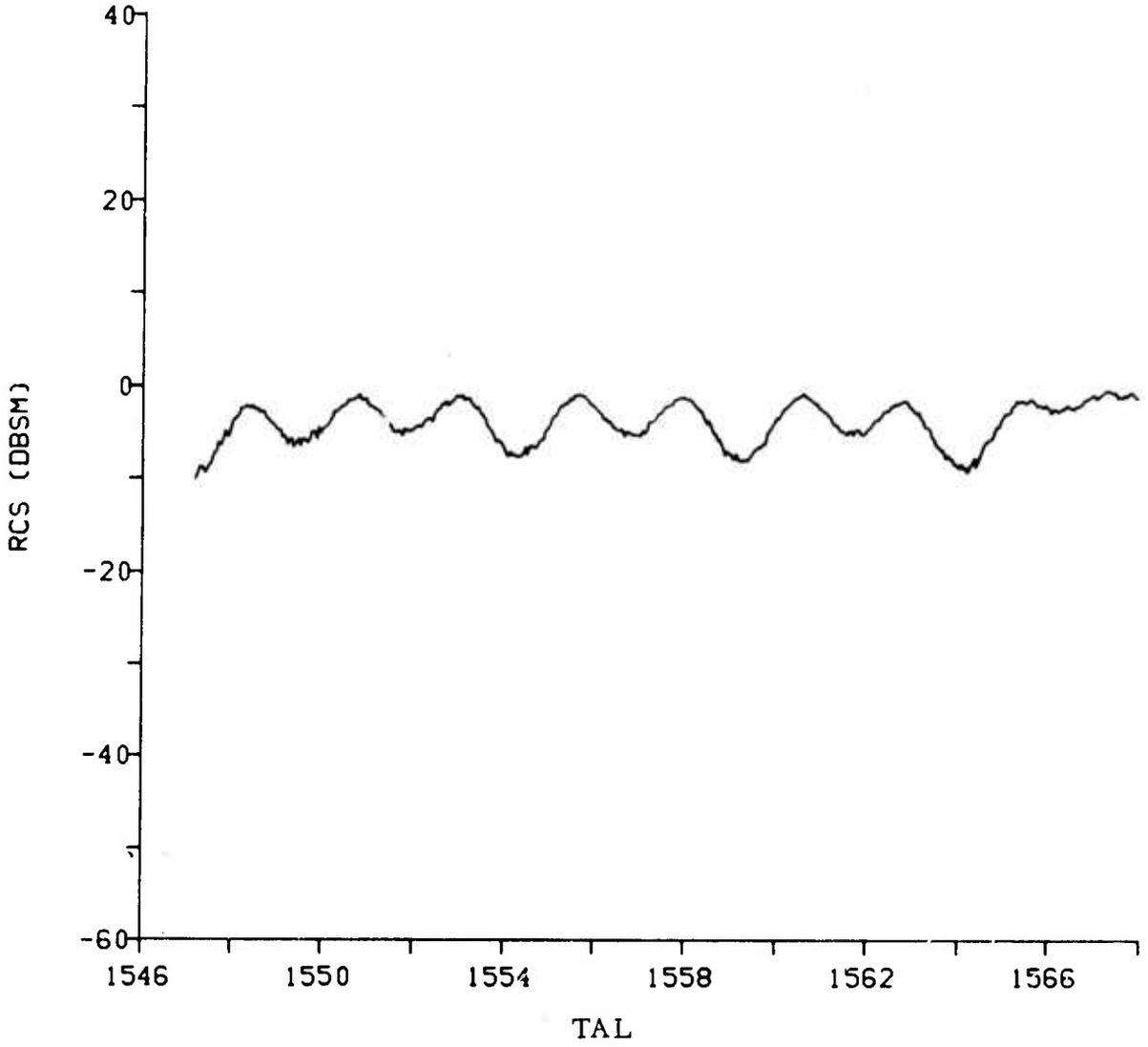
```
ALTOAK EGT AL                                                                    CARD 1
01 43 44.00  01 44 24.00          5    7    1    2     0.05      0.0      1    1
                                                                                 CARD 2
   70   11   04                                                                 CARD 3
```

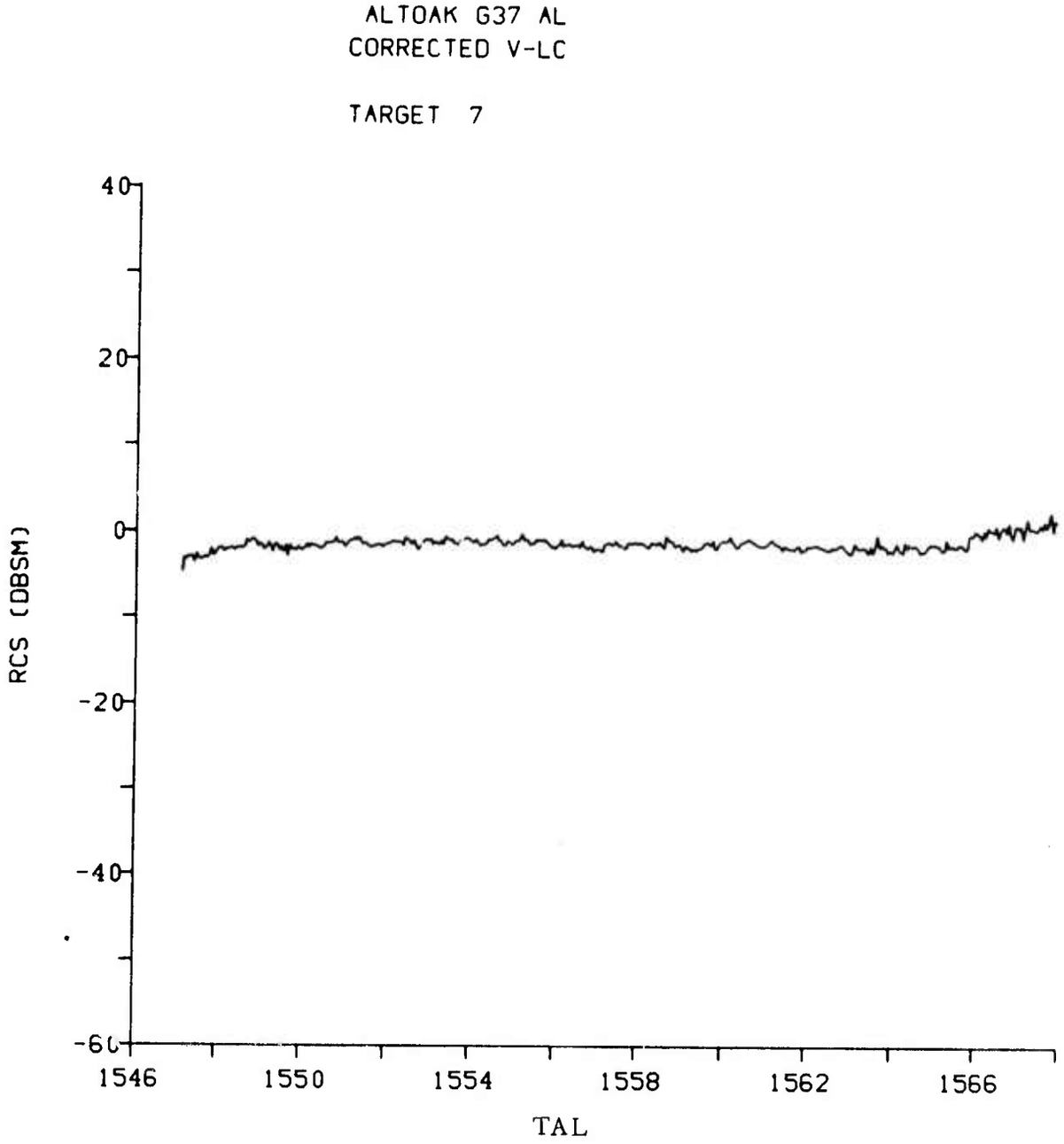
APPENDIX B
ALTOAK OUTPUT

```
ALTAIR OAK VERSION  27 JAN 1971

 ALTOAK 037 A1
TARGET NUMBER =      7


                    AT TIME =     6224.0500  THERE IS A TIME  GAP OR BAD DATA   EXTEN =      0.0000

                    AT TIME =     6224.1000  THERE IS A TIME  GAP OR BAD DATA   EXTEN =      0.0000
                                                 DELTA  DELTA   DELTA   DELTA   DELTA
      TIME     AZ      EL   V-LC(DB) AZ(DB) EL(DB) AZ DEG EL DEG DEGREES VHF(DB) UHF(DB)    PHAZ      PHEL      CRLC
1 44  3.225   50.10   6.08 -2.58   -11.64 -14.03  -0.70   0.55    0.89    2.36   15.97   148.057    80.785   -0.219
1 44  3.275   50.11   6.07 -2.21   -11.46  -9.83  -0.68  -C.80    1.04    3.12   27.87   140.533   113.979    0.917
1 44  3.325   50.10   6.07 -2.40   -10.71 -12.27  -0.76   0.65    0.99    2.87   22.79   151.160    85.237    0.472
1 44  3.374   50.09   6.06 -1.87   -11.85 -12.17  -0.62   C.62    0.88    2.34   15.77   154.565    70.043    0.463
1 44  3.425   50.09   6.06 -2.28   -11.81 -12.14  -0.66   C.65    0.92    2.54   17.20   150.022    76.300    0.269
1 44  3.475   50.07   6.05 -2.17   -11.31  -9.79  -0.69  -0.80    1.05    3.16   28.53   149.106    90.296    0.982
1 44  3.525   50.05   6.04 -2.48   -12.41 -13.76  -0.63   C.56    0.84    2.13   14.47   151.383    86.251   -C.349
1 44  3.575   50.03   6.04 -2.16   -12.92 -11.53  -0.59  -0.68    0.90    2.44   16.46   154.176    91.615    0.276
1 44  3.624   50.00   6.03 -2.17   -11.83  -9.72  -0.65   C.80    1.03    3.04   26.31   144.969    72.433    0.872
1 44  3.675   49.98   6.02 -1.86   -12.21  -9.79  -0.59   C.77    0.97    2.78   20.28   149.739    83.618    0.917
1 44  3.725   49.95   6.02 -1.66   -12.08 -13.42  -0.59   0.53    0.79    1.83   12.93   132.636    73.619    0.186
1 44  3.775   49.92   6.01 -1.25   -12.60  -9.13  -0.50  -C.78    0.93    2.55   17.27   139.412    96.235    1.298
1 44  3.825   49.88   6.01 -1.32   -13.18 -12.69  -0.46   C.56    0.72    1.35   10.40   131.638    86.875    0.027
1 44  3.874   49.84   6.00 -1.11   -14.45 -13.81  -0.35   0.48    0.63    0.62    6.78   153.539    77.303   -0.692
1 44  3.925   49.78   5.99 -0.90   -13.53  -9.66  -0.38  -C.72    0.81    1.97   13.67   132.402    95.574    1.078
1 44  3.975   49.73   5.99 -0.86   -13.85  -9.79  -0.36  -0.71    0.80    1.95   13.01   124.782    90.135    0.998
1 44  4.025   49.68   5.98 -1.29   -15.49  -9.57  -0.33  -0.75    0.82    2.00   13.78   120.969    97.514    0.707
1 44  4.074   49.64   5.98 -1.07   -15.61  -9.18  -0.32  -C.76    0.82    2.04   13.99   115.547   103.419    0.966
1 44  4.124   49.53   5.97 -0.87   -16.76 -13.63  -0.27   0.48    0.55    0.38    6.06   120.528    80.515   -0.492
1 44  4.175   49.47   5.96 -0.53   -15.95  -9.40  -0.29  -0.71    0.77    1.67   12.09   100.863   104.719    1.141
1 44  4.225   49.40   5.95 -0.39   -14.41  -9.45  -0.33  -C.77    0.83    2.10   14.31   109.812   102.722    1.707
1 44  4.275   49.33   5.95 -0.36   -12.37 -13.02  -0.44  -0.48    0.65    0.96    8.22    92.064    91.142    0.632
1 44  4.326   49.26   5.94 -0.59   -14.52 -10.80  -0.34  -C.63    0.71    1.31   10.18   101.028   105.175    0.714
1 44  4.375   49.22   5.93 -0.66   -14.06 -10.87  -0.38  -C.63    0.72    1.36   10.44    79.142    92.521    0.731
1 44  4.425   49.14   5.92 -0.69   -14.07 -12.03  -0.35  -0.55    0.65    0.92    7.99    73.961    95.238    0.428
1 44  4.475   49.07   5.91 -1.18   -12.16 -12.48  -0.54  -0.56    0.77    1.71   12.30    69.939   118.268    0.827
1 44  4.525   49.01   5.90 -0.97   -10.90 -12.37  -0.63  -0.55    0.83    2.10   14.31    81.464   139.266    1.131
1 44  4.576   48.97   5.89 -1.14   -12.66 -11.18  -0.49  -0.64    0.80    1.99   13.23    60.722    91.110    0.750
1 44  4.626   48.92   5.88 -0.84   -11.24 -10.74  -0.59  -C.65    0.87    2.31   15.61    66.992   105.811    1.471
1 44  4.675   48.88   5.87 -0.98   -11.71 -12.09  -0.56  -0.57    0.80    1.87   13.10    62.121    94.341    0.888
1 44  4.725   48.85   5.86 -0.72   -11.67 -10.85  -0.54  -C.63    0.83    2.07   14.18    48.513   118.856    1.398
1 44  4.776   48.80   5.85 -0.53    -9.73  -8.20  -0.69  -C.79    1.05    3.13   28.01    61.519   116.303    2.597
1 44  4.825   48.76   5.84 -0.87   -11.51 -14.59  -0.57   0.42    0.71    1.28   10.03    65.083    84.292    0.413
1 44  4.875   48.74   5.84 -1.11   -10.70 -10.08  -0.65  -C.71    0.96    2.72   18.49    56.055    97.031    1.603
AT TIME =     6310.0360 TARGET  7 IS NOT ON THE TAPE, TARGETS AVAILABLE ARE  12
1 44  4.925   48.71   5.84 -1.12   -10.63 -10.51  -0.66  -0.68    0.95    2.64   17.92    41.946    99.070    1.527

                                  RERANG     REELV      AZCR
                                 980.276    8.227     50.518
                                 980.135    7.281     50.531
                                 979.863    7.433     50.548
                                 979.555    7.429     50.546
                                 979.250    8.610     50.525
                                 978.945    7.330     50.477
                                 978.638    7.464     50.512
                                 978.296    7.347     50.451
                                 977.989    7.397     50.454   <-- CORRECTED R, EL AND AZ FOR INPUT TO NRTPOD (ALSO OUTPUT ON PUNCHED CARDS)
                                 977.683    8.452     50.477
                                 977.375    7.551     50.447
                                 977.033    7.545     50.588
                                 976.727    7.572     50.468
                                 976.420    7.539     50.540
                                 976.080    7.482     50.524
                                 975.773    7.602     50.433
                                 975.466    7.546     50.428
                                 975.125    7.505     50.497
                                 974.818    7.454     50.552
                                 974.510    7.439     50.425
                                 974.169    7.498     50.536
                                 973.861    7.490     50.514
                                 973.554    7.488     50.447
                                 973.213    7.586     50.453
                                 972.906    7.467     50.345
                                 972.600    7.556     50.253
                                 972.294    7.543     50.537
                                 971.953    7.410     50.513
```

ALTOAK G37 AL
UNCORRECTED V-LC

TARGET 7




ALTOAK G37 AL
CORRECTED V-LC
TARGET 7
40
20
0
-20
-40
-60
RCS (DBSM)
1546
1550
1554
1558
1562
1566
TAL

6224.146 99.900 5.190

RCS CORRECTION DATA

ALT 7011 4 14044.14634 0 50.518 8.227 990.2759

OBSERVATION DATA (NRTPOD)

APPENDIX C

SUBROUTINE OAKOS PROGRAM LISTING

```
      DIMENSION DFPG(2),SUMSX(4,30),IDENT(15),IHM(2),UPLOG(4,30),          OAK00010
     1SUMA(30),SUME(30),PLTAG(4),PLOTAG(4),MSEC(1200),ISEC(1200)           OAK00020
      COMMCN/OAK/TIM(1200),TEM(1200),CRLC(1200),GTMAX(1200),AVGSX(4,30),   OAK00030
     1DEL(30),RRANG(1200),RFLEV(1200),AZCR(1200),DAZ(30),IHRX(1200),       OAK00040
     2IMINX(1200)                                                          OAK00050
      COMMCN/RDCOMT/TIMES(300),XSPHA(4,30,300),RANGKM(300),ALSAV(300),     OAK00060
     1AZI(300),ELE(300),IRGA(30),IPOL(4),NPOL                              OAK00070
      DOUBLE PRECISION AVGTM,SEC,SUMTM,T1,T2,TIMES,TOTIM,TSTART,TSTOP,     OAK00080
     1TSV,ZSEC,ZSEC1,ZSEC2,TIP,TLIFT,TEM,TM2,FRAC                          OAK00090
      TOTIM(IH,IM,SEC)=DFLOAT(60*(60*IH+IM))+SEC                           OAK00100
      DATA IMIL/1000000/                                                   OAK00110
      DATA IORBIT/0/                                                       OAK00120
      DATA RADAR/'ALT'/                                                    OAK00130
      DATA SUME/30*0.0/                                                    OAK00140
      DATA SUMA/30*0.0/                                                    OAK00150
      DATA SUMSX/120*0.0/                                                  OAK00160
      DATA UPLOG/120*0.0/                                                  OAK00170
      DATA PLOTAG/'UNCORRECTED V-LC'/                                      OAK00180
      DATA PLTAG/'CORRECTED V-LC  '/                                       OAK00190
      CALL REREAD(99,530)                                                  OAK00200
      TSTOP=0.0                                                            OAK00210
      IAGAIN=0                                                             OAK00220
   41 SUMTM=0.0                                                            OAK00230
      ICARD=0                                                              OAK00240
      NPTS=0                                                               OAK00250
      INDEX=0                                                              OAK00260
      READ(5,20,END=901)IDENT                                              OAK00270
   20 FORMAT(15A4)                                                         OAK00280
      READ(5,60)IH1,IM1,ZSEC1,IH2,IM2,ZSEC2,NPG,INTARG,IPAT                OAK00290
     1,ISG,TAVG,TSKIP,ICARD,INRT                                           OAK00300
   60 FORMAT(2(2I3,F7.3),4X,4I5,2F10.3,2I5)                                OAK00310
      IF(INRT.GT.0)READ(5,61)IYEAR,IMONTH,IDAY                             OAK00320
   61 FORMAT(3I5)                                                          OAK00330
      IPOL(1)=1                                                            OAK00340
      IPOL(2)=3                                                            OAK00350
      IPOL(3)=4                                                            OAK00360
      IPOL(4)=0                                                            OAK00370
      ISTGT=1                                                              OAK00380
      WRITE(6,64)                                                          OAK00390
   64 FORMAT(5X,'THESE ARE YOUR INPUT CARDS')                              OAK00400
      WRITE(6,62)IH1,IM1,ZSEC1,IH2,IM2,ZSEC2,NRG,INTARG,IPAT,IPOL,         OAK00410
     1ISG,TAVG,TSKIP                                                       OAK00420
   62 FORMAT(2(2I3,F7.3),4X,8I3,2F10.3)                                    OAK00430
      WRITE(6,63)ISTGT                                                     OAK00440
   63 FORMAT(I5)                                                           OAK00450
      CALL STOIDV(IDENT,59,0)                                              OAK00460
      IF(NRG.EQ.0)GO TO 550                                                OAK00470
      IF(INTARG.EQ.0)GO TO 560                                             OAK00480
      IF((IPAT.EQ.0).OR.(IPAT.GT.3))GO TO 570                              OAK00490
      IF(TAVG.EQ.0.0)GO TO 590                                             OAK00500
   70 TSTART=TOTIM(IH1,IM1,ZSEC1)                                          OAK00510
      IF((TSTART.GT.TSTOP).AND.(IAGAIN.NE.44))GO TO 72                     OAK00520
      CALL REW                                                             OAK00530
      IAGAIN=1                                                             OAK00540
   72 TSTOP=TOTIM(IH2,IM2,ZSEC2)                                           OAK00550
```

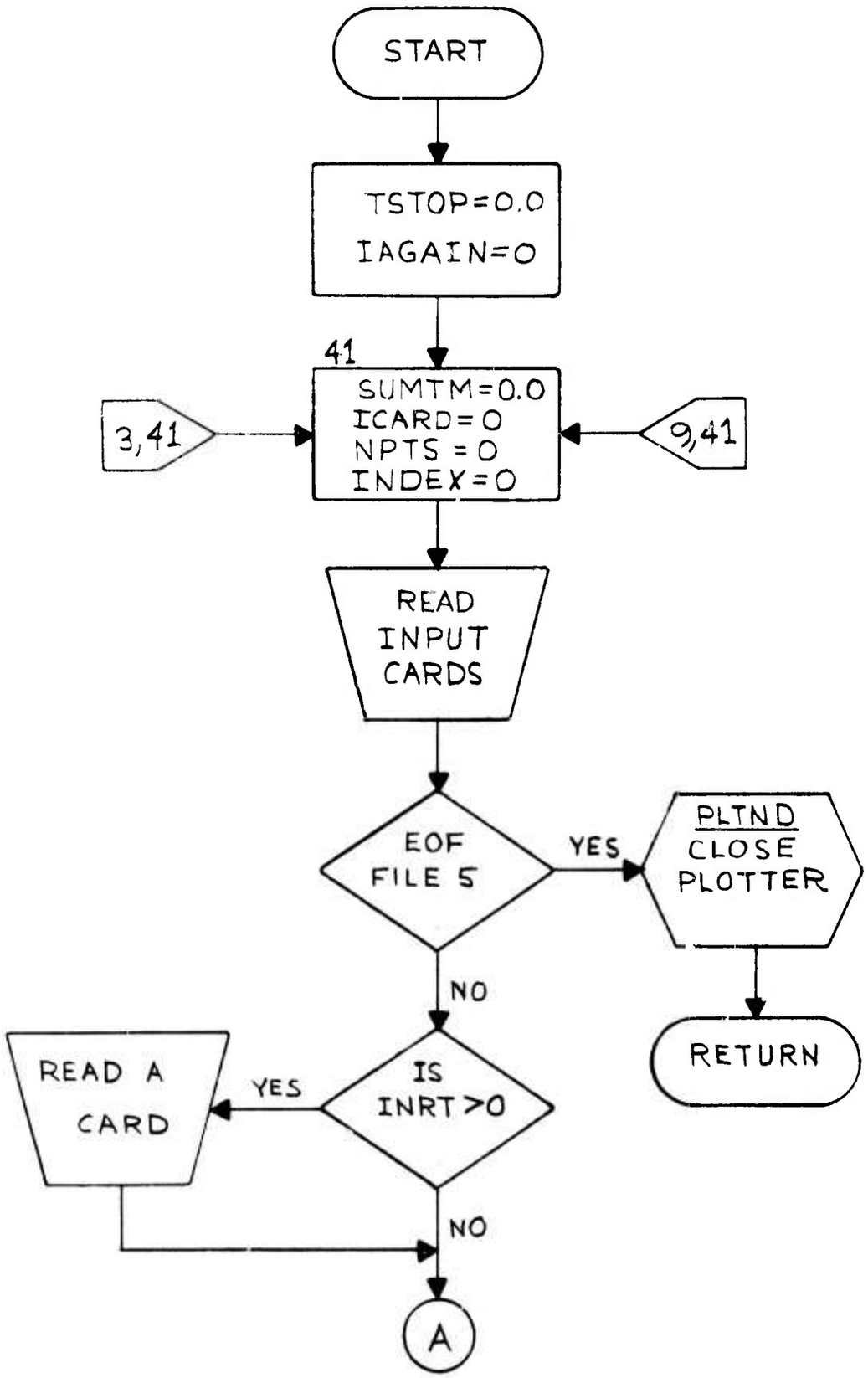
```
      COUNT=0                                                          OAK00560
      INT=0                                                            OAK00570
      T1=TSTART                                                        OAK00580
      T2=T1+TAVG                                                       OAK00590
  100 CALL ALREAD(TSTART,TSTOP,TLIFT,INTARG,IPAT,2,NPTS,NFPG,IAGAIN,   OAK00600
     1NRG,ISG)                                                         OAK00610
      IF(IAGAIN.EQ.55)GO TO 901                                        OAK00620
      IF(IAGAIN.EQ.44)GO TO 510                                        OAK00630
  110 IF(NPTS.EQ.0)GO TO 41                                            OAK00640
      IF(INT.EQ.0)WRITE(6,140)IDENT,INTARG                             OAK00650
  140 FORMAT('1',30X' ALTAIR OAK VERSION  27 JAN 1971       '//31X,    OAK00660
     115A4/31X,'TARGET NUMBER = ',I5//)                                OAK00670
      IF((INT.EQ.0).AND.(ICARD.GT.0))WRITE(7,150)IDENT                 OAK00680
  150 FORMAT(15A4)                                                     OAK00690
      INT=1                                                            OAK00700
      DO 220 I=1,NPTS                                                  OAK00710
  160 IF(TIMES(I).GT.T2)GO TO 240                                      OAK00720
      IF(T1.GT.TIMES(I))GO TO 220                                      OAK00730
      INPT=I                                                           OAK00740
      DO 210 N=1,NPOL                                                  OAK00750
      DO 200 K=1,NRG                                                   OAK00760
      EXTEN=(XSPHA(N,K,I)/10.)                                         OAK00770
      IF(EXTEN.GT.75.0)GO TO 245                                       OAK00780
      XSPHA(N,K,I)=10.**EXTEN                                          OAK00790
      SUMSX(N,K)=SUMSX(N,K)+XSPHA(N,K,I)                               OAK00800
      IF(N.GT.1)GO TO 200                                              OAK00810
  185 XSPHQ=XSPHA(2,K,I+150)-XSPHA(1,K,I+150)                          OAK00820
      IF(ABS(XSPHQ).LT.180.)GO TO 190                                  OAK00830
      IF(XSPHQ.GT.0.0)XSPHA(1,K,I+150)=XSPHA(1,K,I+150)+360.           OAK00840
      IF(XSPHQ.LT.0.0)XSPHA(2,K,I+150)=XSPHA(2,K,I+150)+360.           OAK00850
      GO TO 185                                                        OAK00860
  190 XSPHQ=ABS(XSPHQ)                                                 OAK00870
      SUMA(K)=SUMA(K)+XSPHQ                                            OAK00880
  191 XSPHX=XSPHA(3,K,I+150)-XSPHA(1,K,I+150)                          OAK00890
      IF(ABS(XSPHX).LT.180.)GO TO 192                                  OAK00900
      IF(XSPHX.GT.0.0)XSPHA(1,K,I+150)=XSPHA(1,K,I+150)+360.           OAK00910
      IF(XSPHX.LT.0.0)XSPHA(3,K,I+150)=XSPHA(3,K,I+150)+360.           OAK00920
      GO TO 191                                                        OAK00930
  192 XSPHX=ABS(XSPHX)                                                 OAK00940
      SUME(K)=SUME(K)+XSPHX                                            OAK00950
  200 CONTINUE                                                         OAK00960
  210 CONTINUE                                                         OAK00970
      COUNT=COUNT+1                                                    OAK00980
  220 CONTINUE                                                         OAK00990
      IF(IAGAIN.EQ.0)GO TO 240                                         OAK01000
      NPTS=0                                                           OAK01010
      GO TO 100                                                        OAK01020
  240 IF(COUNT.NE.0.0)GO TO 280                                        OAK01030
  245 WRITE(6,260)T2,EXTEN                                             OAK01040
  260 FORMAT(/25X'AT TIME = 'F12.4,2X'THERE IS A TIME  GAP OR BAD DATA OAK01050
     1 EXTEN = ',F10.4)                                                OAK01060
      GO TO 440                                                        OAK01070
  280 DO 350 M=1,NPOL                                                  OAK01080
      DO 340 J=1,NRG                                                   OAK01090
      IF(M.GT.1)GO TO 290                                              OAK01100
```

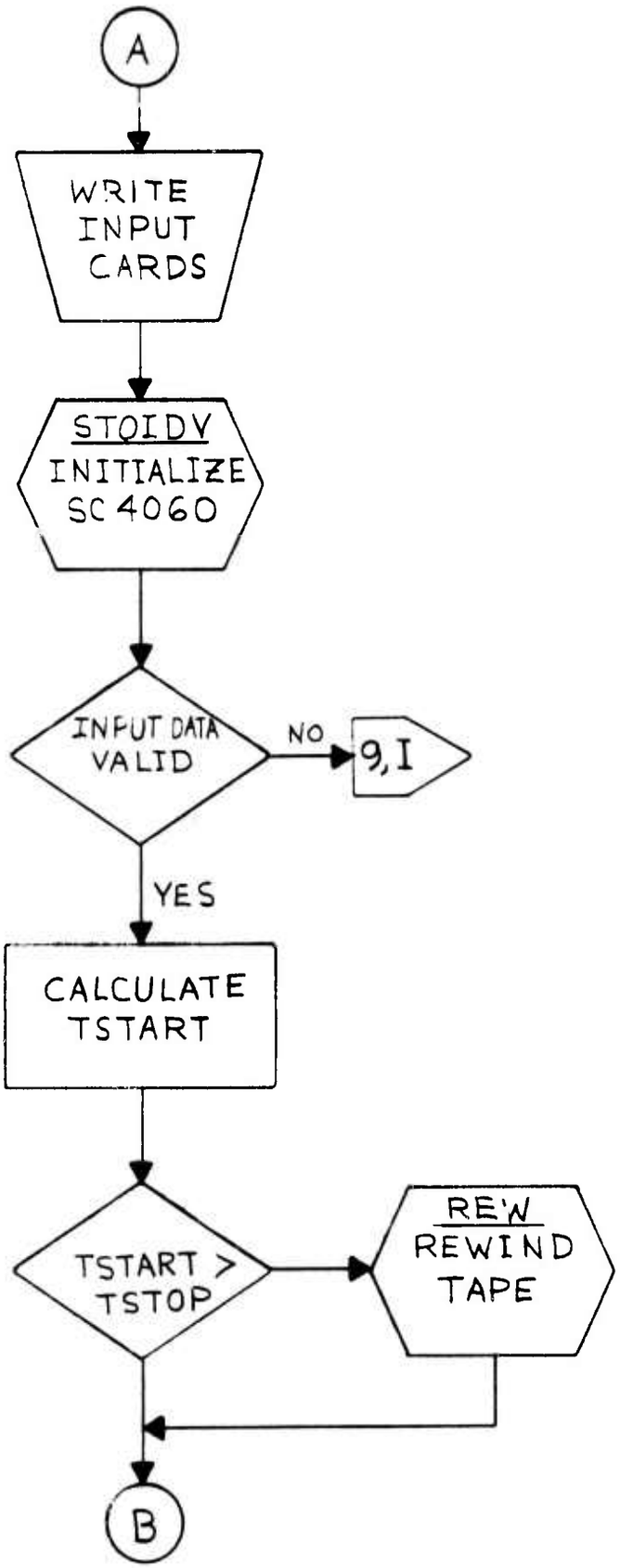
```
      DAZ(J)=SUMA(J)/COUNT                                                    OAK01110
      DEL(J)=SUME(J)/COUNT                                                    OAK01120
  290 UPLOG(M,J)=SUMSX(M,J)/CCUNT                                             OAK01130
      IF(UPLOG(M,J).GT.0.0) GO TO 300                                         OAK01140
      AVGSX(M,J)=99.9                                                         OAK01150
      GO TO 320                                                               OAK01160
  300 AVGSX(M,J)=10.*ALOG10(UPLOG(M,J))                                       OAK01170
  320 SUMSX(M,J)=0.0                                                          OAK01180
      SUMA(J)=0.0                                                             OAK01190
      SUME(J)=0.0                                                             OAK01200
  340 CONTINUE                                                                OAK01210
  350 CONTINUE                                                                OAK01220
      MIDPT=((COUNT+1)/2)+INPT                                                OAK01230
      AVGEL=ELE(MIDPT)                                                        OAK01240
      AVGAZ=AZI(MIDPT)                                                        OAK01250
      AVGTM=TIMES(MIDPT)                                                      OAK01260
      AVGRG=RANGKM(MIDPT)                                                     OAK01270
      AVGAL=ALSAV(MIDPT)                                                      OAK01280
      KOUNT=COUNT                                                             OAK01290
  358 IF(ISTGT.EQ.0) GO TO 440                                                OAK01300
      CALL GLMP(AVGAL,ISTGT,AVGAZ,AVGEL,IGAT,AVGTM,AVGRG,NRG,INDEX,          OAK01310
     1ICARD,TLIFT)                                                            OAK01320
  440 COUNT=0                                                                 OAK01330
      T1=T2+TSKIP                                                             OAK01340
      T2=T1+TAVG                                                              OAK01350
      SUMTM=0.0                                                               OAK01360
      SUMRG=0.0                                                               OAK01370
      SUMAL=0.0                                                               OAK01380
      IF(T2.LE.TIMES(NPTS)) GO TO 160                                         OAK01390
      IF(T2.GT.TSTOP) GO TO 510                                               OAK01400
      DO 460 K=INPT,NPTS                                                      OAK01410
      KNPT=K                                                                  OAK01420
      IF(T1.LE.TIMES(K)) GO TO 480                                            OAK01430
  460 CONTINUE                                                                OAK01440
  480 ND=NPTS-KNPT+1                                                          OAK01450
      DO 501 N=1,ND                                                           OAK01460
      NL=KNPT+N-1                                                             OAK01470
      TIMES(N)=TIMES(NL)                                                      OAK01480
      RANGKM(N)=RANGKM(NL)                                                    OAK01490
      ALSAV(N)=ALSAV(NL)                                                      OAK01500
      AZI(N)=AZI(NL)                                                          OAK01510
      ELE(N)=ELE(NL)                                                          OAK01520
      DO 490 K=1,NPOL                                                         OAK01530
      DO 500 L=1,NRG                                                          OAK01540
      XSPHA(K,L,N)=XSPHA(K,L,NL)                                              OAK01550
  500 CONTINUE                                                                OAK01560
  490 CONTINUE                                                                OAK01570
  501 CONTINUE                                                                OAK01580
      NPTS=ND                                                                 OAK01590
      IF(IAGAIN.NE.0) GO TO 100                                               OAK01600
  510 CALL OAKPLT(TIM,GTMAX,INDEX,PLOTAG,INTARG,IDENT)                        OAK01610
      CALL OAKPLT(TIM,CRLC,INDEX,PLTAG,INTARG,IDENT)                          OAK01620
      WRITE(6,540)                                                            OAK01630
  540 FORMAT(44X'RFRANG      RFELV      AZCR')                               OAK01640
      WRITE(6,541) (PRANG(N),RELEV(N),AZCR(N),N=1,INDEX)                      OAK01650
```

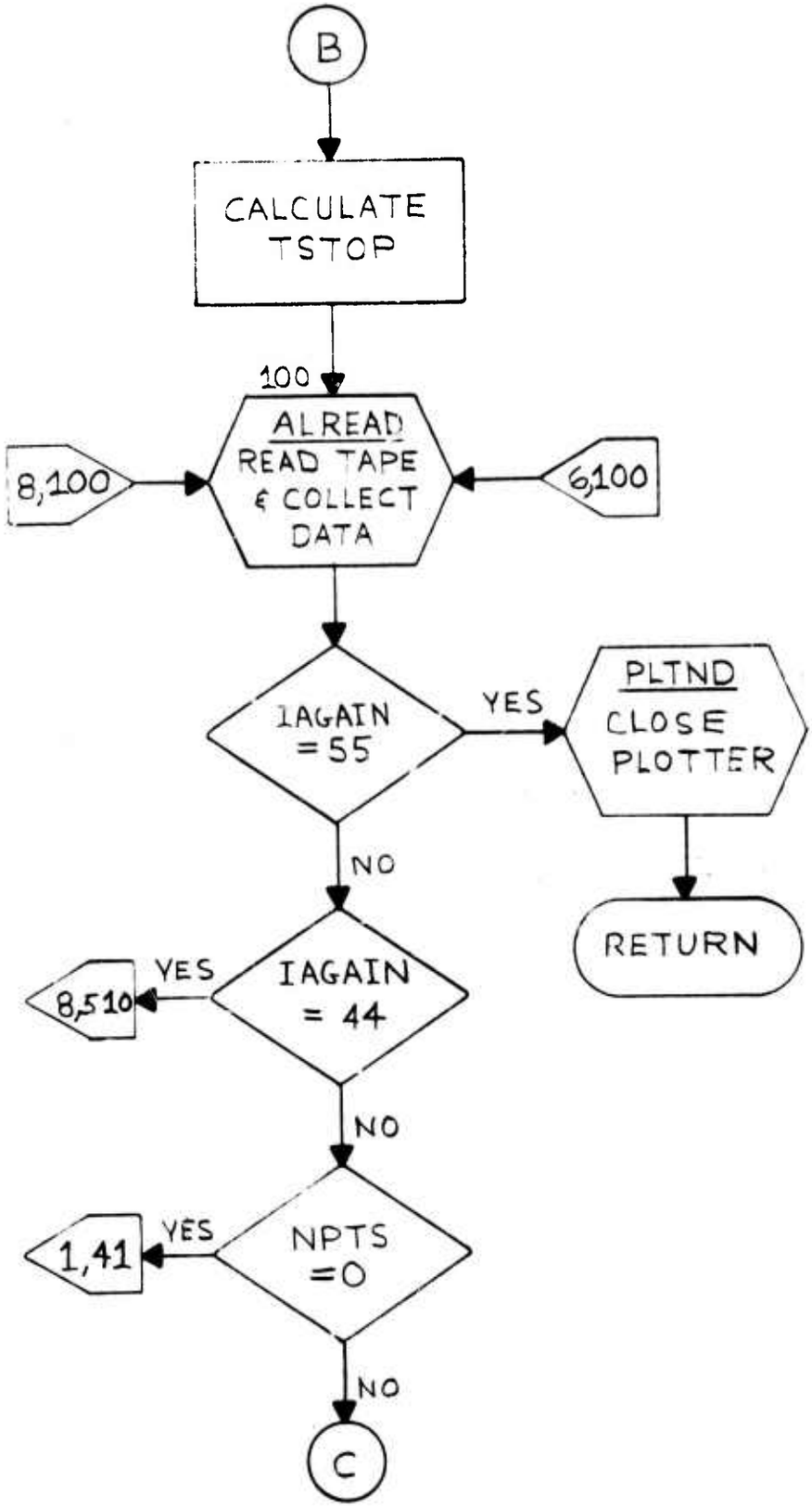
```
 541  FORMAT(40X,3F10.3)                                                    OAK01660
      IF(INRT.EQ.0)GO TO 5411                                               OAK01670
      DO 54100 M=1,INDEX                                                    OAK01680
      TM2=IDINT(TEM(M))                                                     OAK01690
      ISEC(M)=TM2                                                           OAK01700
      FRAC=TEM(M)-TM2                                                       OAK01710
      IFRAC=IDINT(FRAC*IMIL)                                                OAK01720
      MSEC(M)=DFLOAT(IFRAC)/10.+.5                                          OAK01730
54100 CONTINUE                                                              OAK01740
      WRITE(7,5410)(RADAR,IYEAR,IMONTH,IDAY,IHRX(I),IMINX(I),ISEC(I),       OAK01750
     1MSEC(I),IORBIT,AZCR(I),RELEV(I),RRANG(I),I=1,INDEX)                   OAK01760
5410 FORMAT(A3,3X,6I2,'.',I5,1X,I1,F8.3,4X,F8.3,4X,F12.4)                   OAK01770
5411 IF(ICARD.EQ.0)GO TO 900                                                OAK01780
      WRITE(6,542)INDEX                                                     OAK01790
 542  FORMAT(///I5,'CORRECTION CARDS HAVE BEEN PUNCHED')                    OAK01800
      GO TO 900                                                             OAK01810
 550  WRITE(6,555)                                                          OAK01820
 555  FORMAT(//2X'A ZERO VALUE WAS INPUT FOR NRG THIS IS A NO NO')          OAK01830
      GO TO 900                                                             OAK01840
 560  WRITE(6,565)                                                          OAK01850
 565  FORMAT(//2X'A ZERO VALUE WAS INPUT FOR THE TARGET #, THE DATA EDITOAK01860
     1OR HAS GOOFED AGAIN')                                                 OAK01870
      GO TO 900                                                             OAK01880
 570  WRITE(6,575)IPAT                                                      OAK01890
 575  FORMAT(//2X'A VALUE OF',I5,'  WAS INPUT FOR IPAT THE ONLY LEGAL VAOAK01900
     1LUES FOR IPAT ARE 1,2,3')                                             OAK01910
      GO TO 900                                                             OAK01920
 590  WRITE(6,595)                                                          OAK01930
 595  FORMAT(//2X'A ZERO VALUE FOR TINC CAN NOT WORK IT WILL BE SET TO      OAK01940
     10.05 SECONDS AND THE PROGRAM WILL CONTINUE')                          OAK01950
      TAVG=0.05                                                             OAK01960
      GO TO 70                                                              OAK01970
 900  IAGAIN=99                                                             OAK01980
      GO TO 41                                                              OAK01990
 901  CALL PLTND                                                            OAK02000
      RETURN                                                                OAK02010
      END                                                                   OAK02020
```

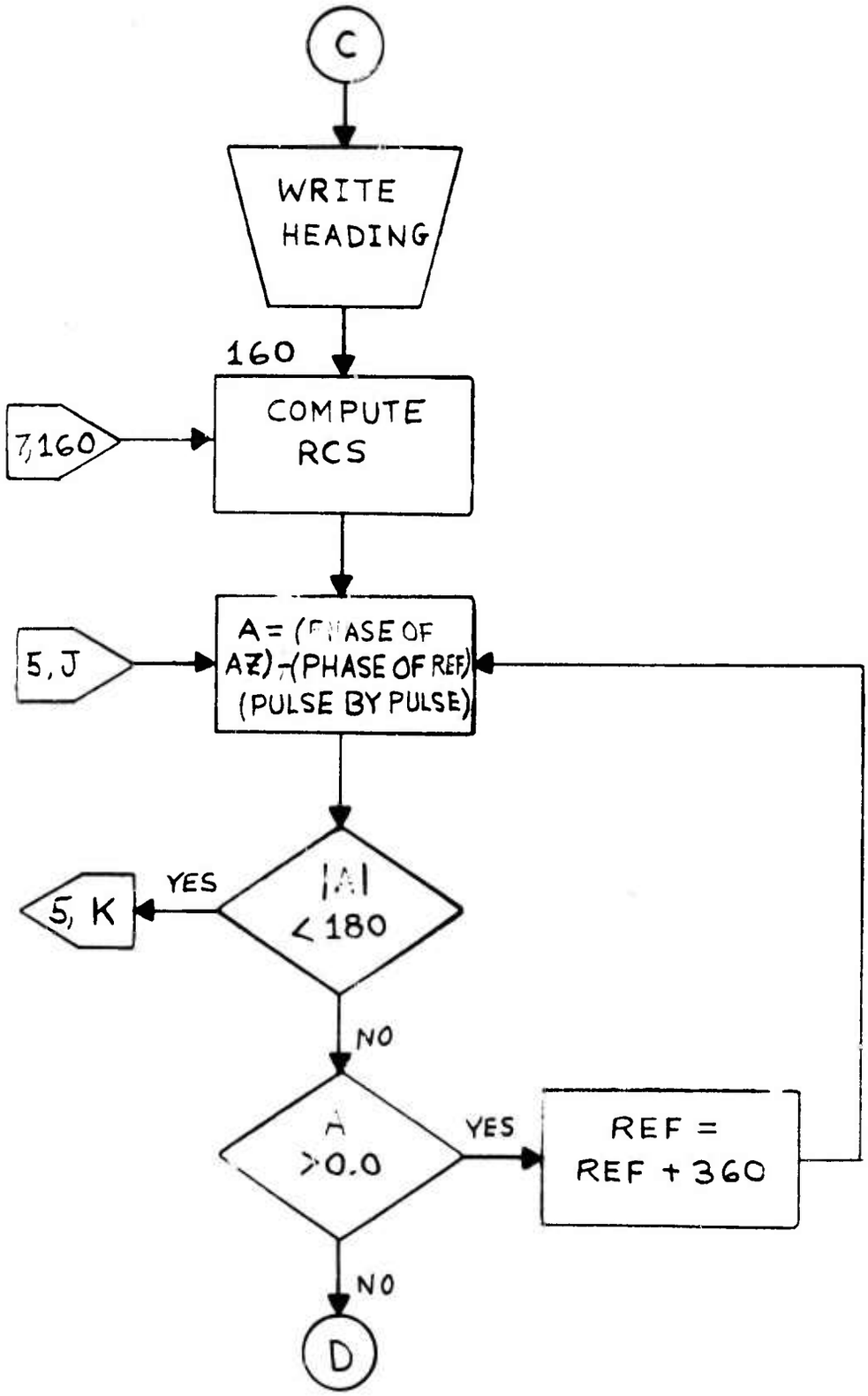
APPENDIX D

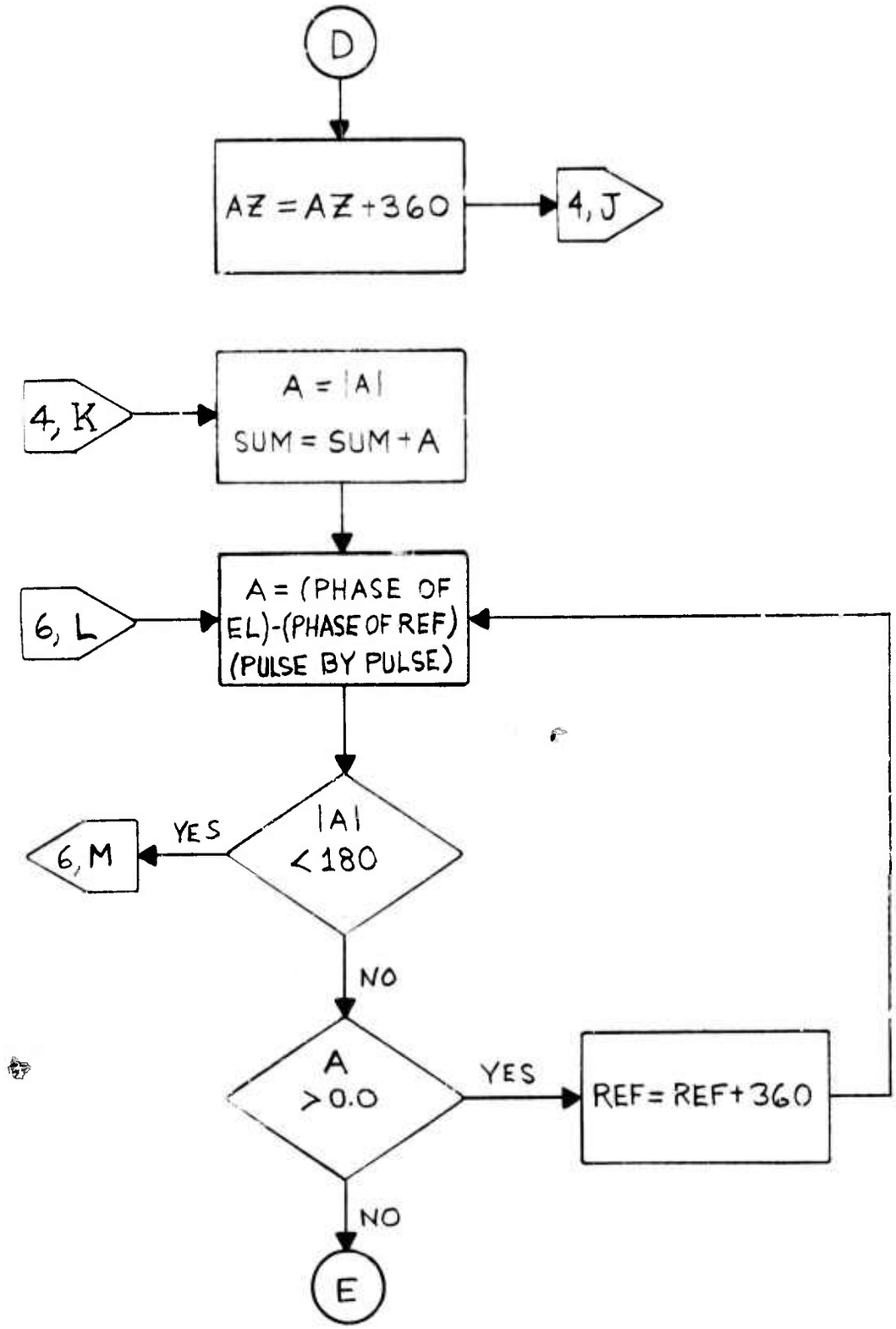
SUBROUTINE OAKOS FLOW DIAGRAM

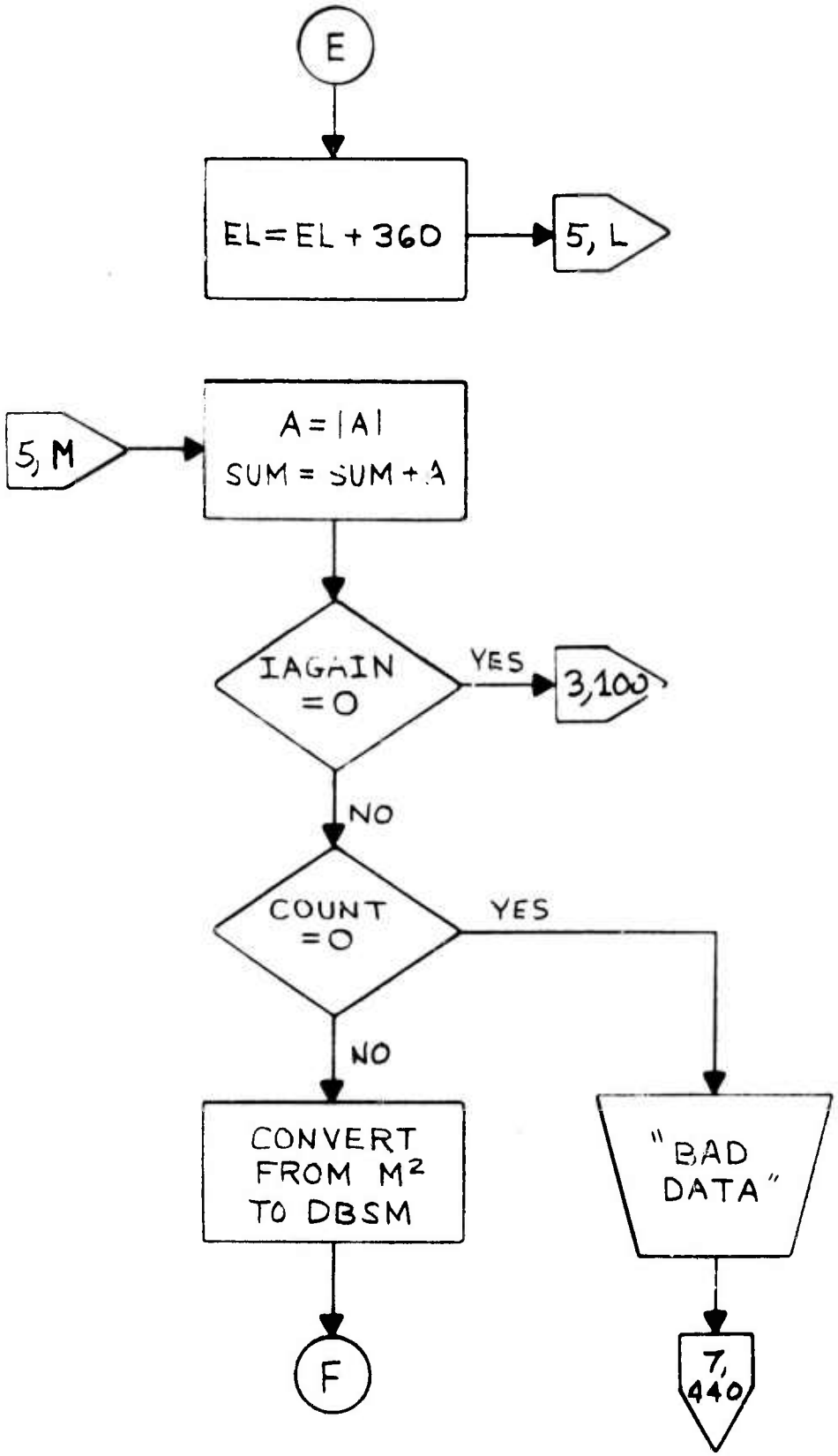


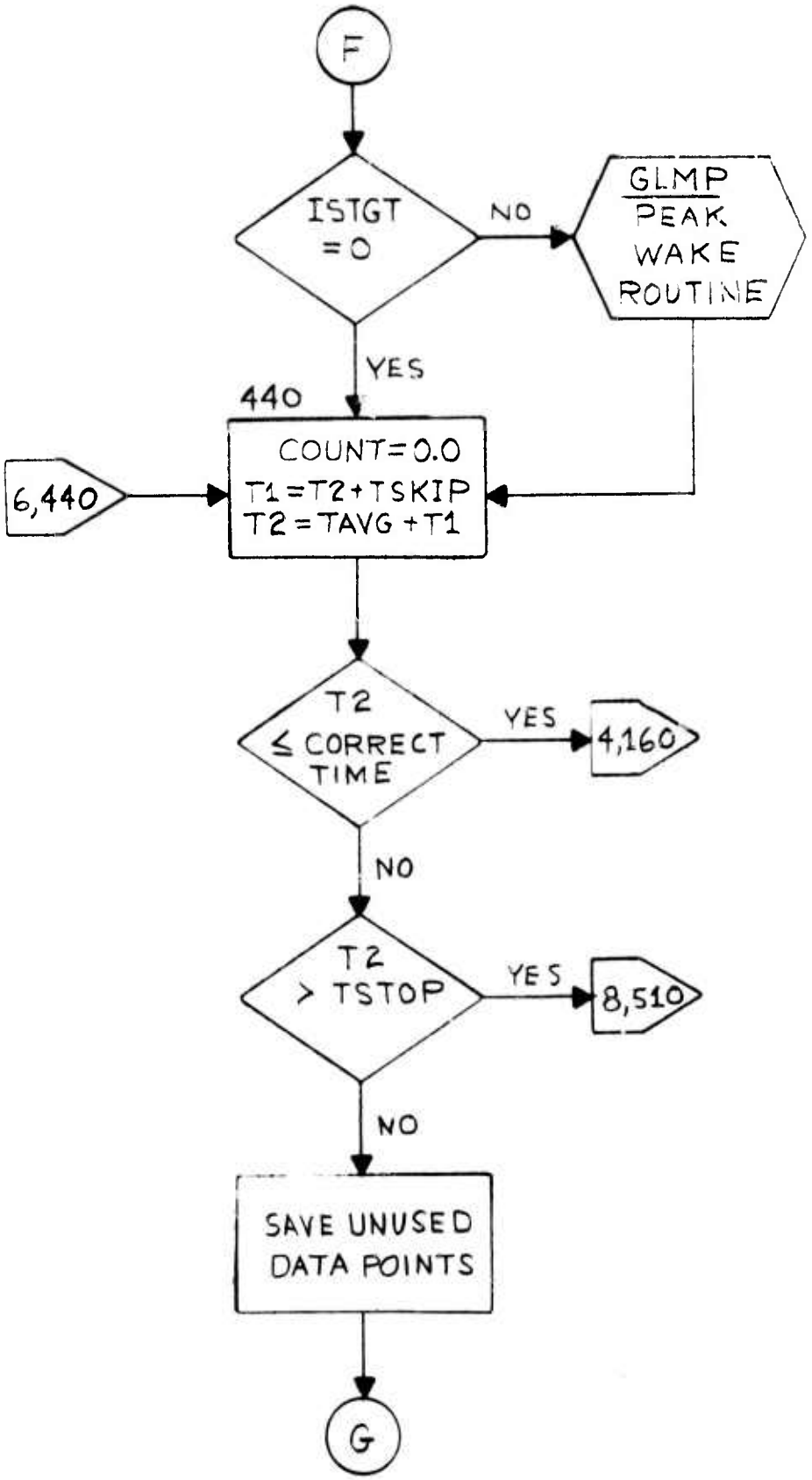

A
WRITE INPUT CARDS
STOIDV INITIALIZE SC 4060
INPUT DATA VALID
NO
9, I
YES
CALCULATE TSTART
TSTART > TSTOP
REW REWIND TAPE
B

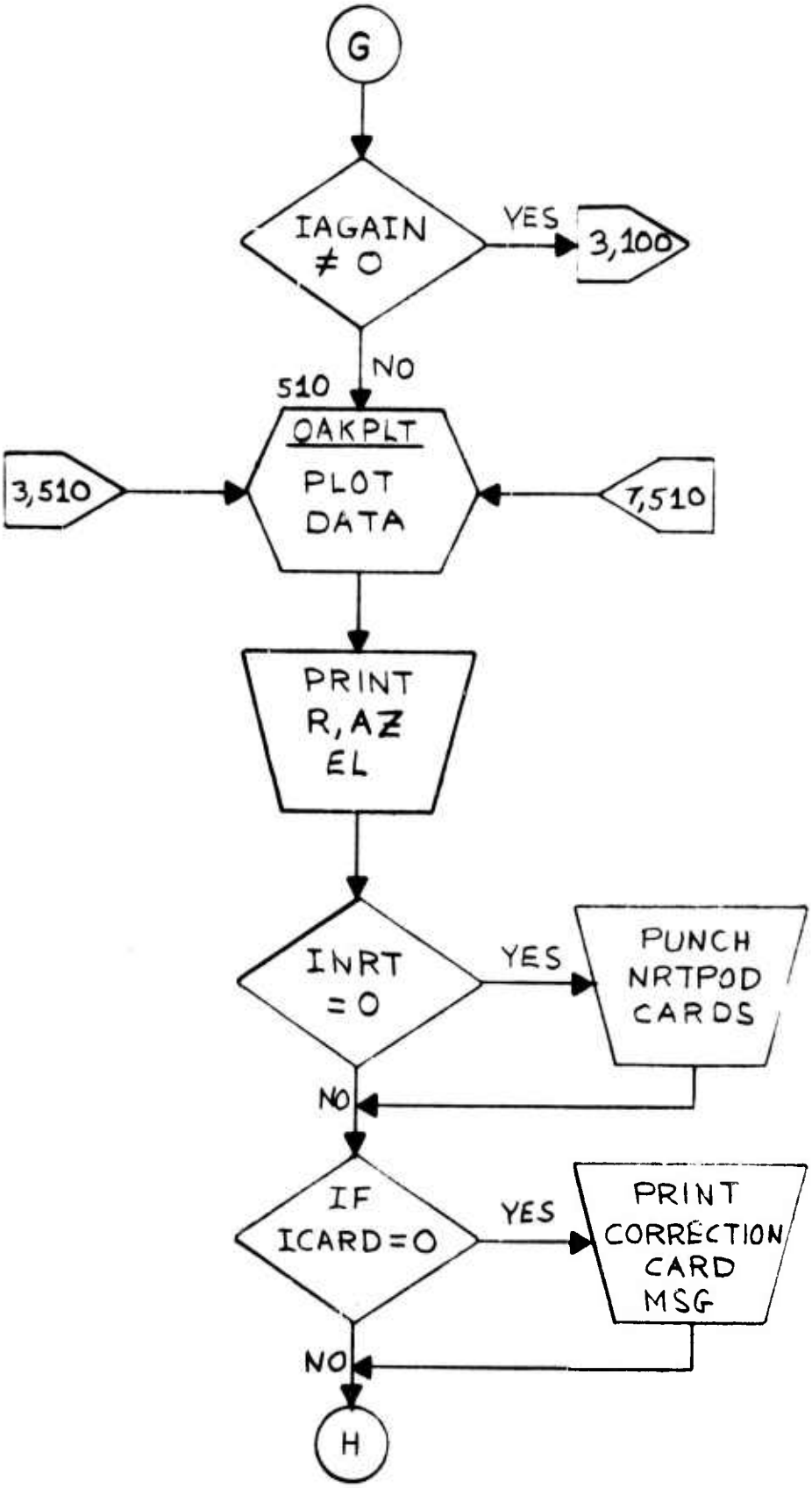

B
CALCULATE TSTOP
100
ALREAD
READ TAPE & COLLECT DATA
8,100
6,100
IAGAIN = 55
YES
PLTND
CLOSE PLOTTER
RETURN
NO
IAGAIN = 44
YES
8,510
NO
NPTS = 0
YES
1,41
NO
C

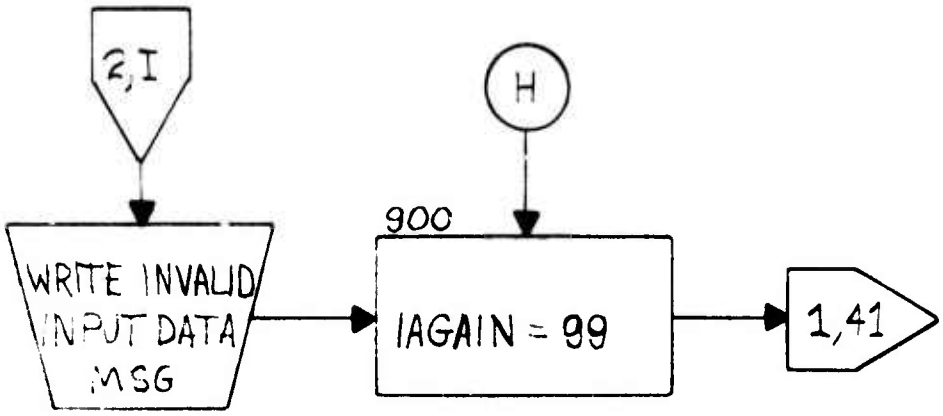

C
WRITE HEADING
160
COMPUTE RCS
7,160
A = (PHASE OF AZ) - (PHASE OF REF)
(PULSE BY PULSE)
5, J
|A| < 180
YES
5, K
NO
A > 0.0
YES
REF = REF + 360
NO
D


D
AZ = AZ + 360
4, J
4, K
A = |A|
SUM = SUM + A
6, L
A = (PHASE OF EL) - (PHASE OF REF) (PULSE BY PULSE)
|A| < 180
YES
6, M
NO
A > 0.0
YES
REF = REF + 360
NO
E


E
EL=EL+360
5, L
5, M
A=|A|
SUM = SUM + A
IAGAIN =0
YES
3,100
NO
COUNT =0
YES
NO
CONVERT FROM M² TO DBSM
"BAD DATA"
F
7, 440


F
ISTGT
=0
NO
GLMP
PEAK
WAKE
ROUTINE
YES
440
COUNT=0.0
T1=T2+TSKIP
T2=TAVG+T1
6,440
T2
≤ CORRECT
TIME
YES
4,160
NO
T2
> TSTOP
YES
8,510
NO
SAVE UNUSED
DATA POINTS
G


G
IAGAIN ≠ 0
YES
3,100
NO
510
OAKPLT
PLOT DATA
3,510
7,510
PRINT R, AZ EL
INRT = 0
YES
PUNCH NRTPOD CARDS
NO
IF ICARD=0
YES
PRINT CORRECTION CARD MSG
NO
H

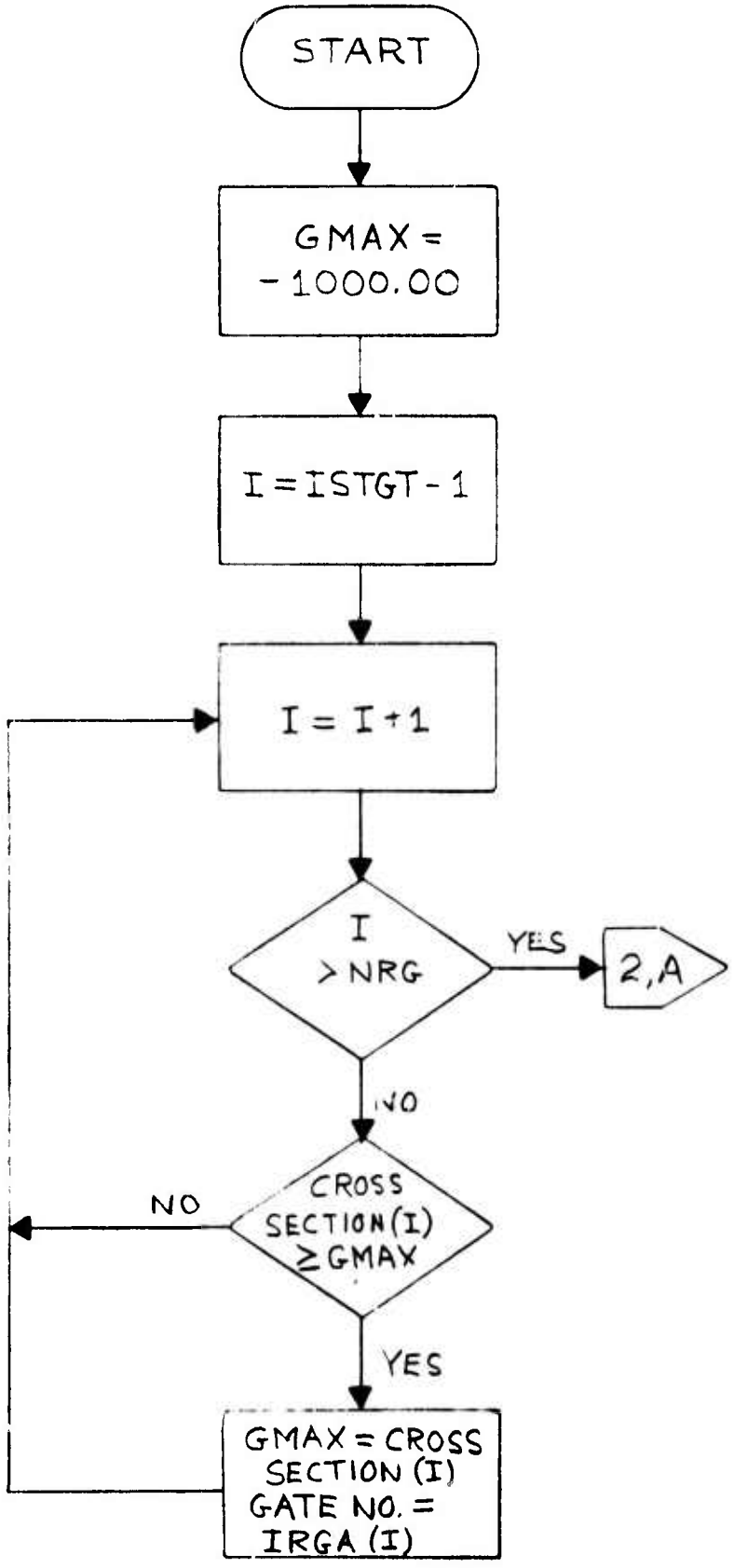

2,I
H
900
WRITE INVALID
INPUT DATA
MSG
IAGAIN = 99
1,41

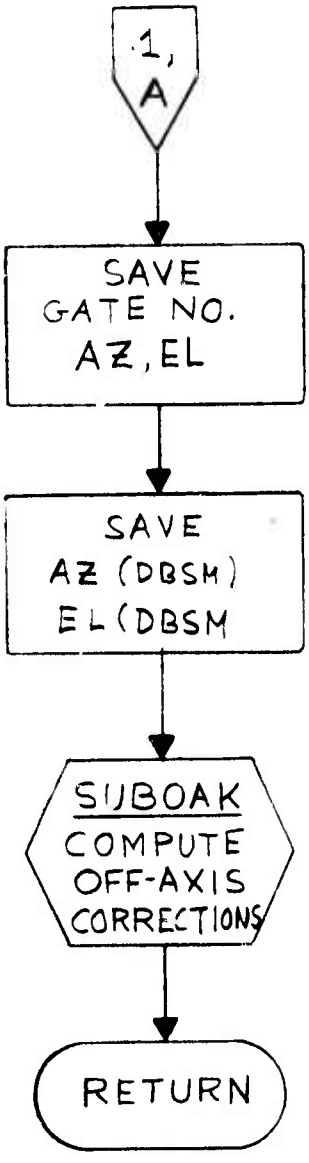
APPENDIX E
SUBROUTINE GLMP PROGRAM LISTING

```
      SUBROUTINE GLMP(AVGAL,ISTGT,AVGAZ,AVGEL,IGAT,AVGTM,AVGRG,NRG      PEAC0010
     1,INDEX,ICARD,TLIFT)
      DIMENSION IHM(2)                                                  PEAC0050
      DOUBLE PRECISION AVGTM,TIMES,ZSEC,TIM,TEM,TLIFT
      COMMON/OAK/TIM(1200),TEM(1200),CRLC(1200),GTMAX(1200),AVGSX(4,30),
     1DEL(30),RRANG(1200),RELEV(1200),AZCR(1200),DAZ(30),IHRX(1200),
     2IMINX(1200)
      COMMON/RDCOMT/TIMES(300),XSPHA(4,30,300),RANGKM(300),ALSAV(300),  PEA00060
     1AZI(300),ELE(300),IRGA(30),IPOL(4),NPOL                           PEA00070
      EQUIVALENCE(IHM(1),IHR),(IHM(2),IMIN)                             PEAC0080
      GMAX=-1000.0                                                      PEA00090
      DO 20 I=ISTGT,NRG                                                 PEA00100
      IF(AVGSX(1,I).LE.GMAX)GO TO 20                                    PEA00110
      GMAX=AVGSX(1,I)                                                   PEA00120
      IGAT=IRGA(I)                                                      PEA00130
      NN=I                                                              PEA00140
20    CONTINUE                                                          PEA00150
      CALL TSPLIT(AVGTM,IHM,ZSEC)                                       PEA00160
      PHAZ=DAZ(NN)                                                      PEA00170
      PHEL=DEL(NN)                                                      PEA00180
      AZ=AVGSX(2,NN)                                                    PEA00190
      EL=AVGSX(3,NN)                                                    PEA00200
      CALL SUBOAK(IHR,IMIN,ZSEC,GMAX,AVGAL,IGAT,AVGTM,AVGAZ,AVGEL,PHAZ,PPEA00210
     1HEL,AZ,EL,INDEX,AVGRG,ICARD,TLIFT)
      RETURN                                                            PEA00230
      END
```

APPENDIX F

SUBROUTINE GLMP FLOW DIAGRAM




1,
A
SAVE
GATE NO.
AZ, EL
SAVE
AZ (DBSM)
EL (DBSM
SUBOAK
COMPUTE
OFF-AXIS
CORRECTIONS
RETURN

APPENDIX G
SUBROUTINE TSPLIT PROGRAM LISTING

```
      SUBROUTINE TSPLIT(AVGTM,IHM,TRUN)                          TSPC0010
      DIMENSION IHM(2),DIVIDE(2)                                 TSP00020
      DOUBLE PRECISION AVGTM,TRUN                                TSPC0030
      DATA DIVIDE/3600.,60./                                     TSP00040
      TRUN=AVGTM                                                 TSPC0050
      DO 20 I=1,2                                                TSP00060
      IHM(I)=TRUN/DIVIDE(I)                                      TSP00070
      TRUN=TRUN-FLOAT(IHM(I))*DIVIDE(I)                          TSPC0080
20    CONTINUE                                                   TSP00090
      RETURN                                                     TSP00100
      END                                                        TSP00110
```

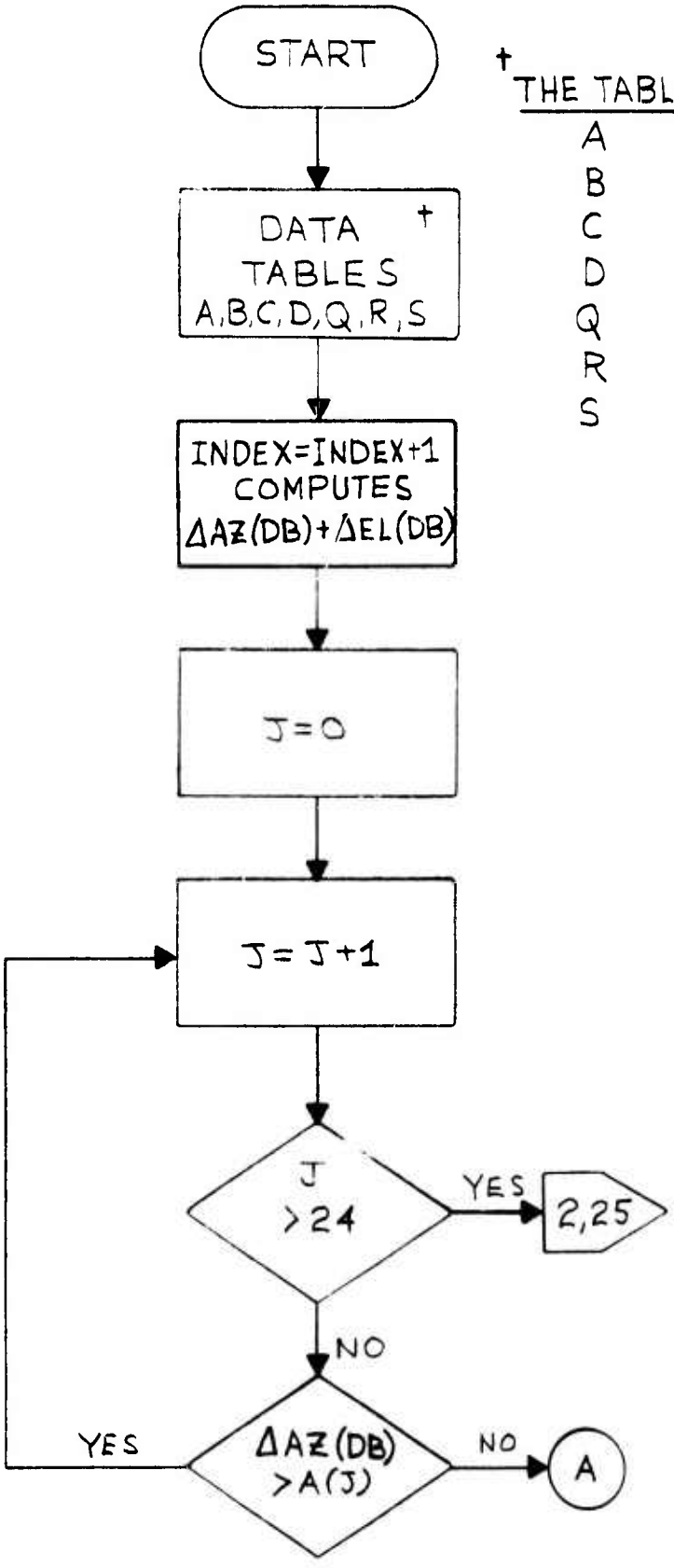
APPENDIX H
SUBROUTINE SUBOAK PROGRAM LISTING

```
      SUBROUTINE SUBOAK(IHR,IMIN,ZSEC,GMAX,AVGAL,IGAT,AVGTM,AZE,ELEX,    SUB00010
     1PHAZ,PHEL,AZ,EL,INDEX,AVGRG,ICARD,TLIFT)
      DIMENSION A(24),B(24),C(27),D(27),Q(27),R(16),S(16)
      DOUBLE PRECISION ZSEC,TIM,AVGTM,TEM,TLIFT
      COMMON/OAK/TIM(1200),TEM(1200),CRLC(1200),GTMAX(1200),AVGSX(4,30),
     1DEL(30),RRANG(1200),RELEV(1200),AZCR(1200),DAZ(30),IHRX(1200),
     2IMINX(1200)
      DATA DR,NIT,NNIT,NITT/.0174533,24,27,27/
      DATA XKMFT/.0003048/
      DATA A/-29.6,-27.3,-23.2,-20.1,-17.0,-12.8,-10.4,-8.4,-6.2,-4.6,
     1-3.2,-1.7,-0.7,1.1,1.7,2.9,4.1,5.1,7.0,8.5,9.5,10.3,10.8,12.4/
      DATA B/0.0,0.03,0.1,0.17,0.24,0.37,0.59,0.75,0.90,1.05,1.26,1.34,
     11.48,1.61,1.75,1.88,2.01,2.14,2.28,2.39,2.61,2.74,2.81,2.93/
      DATA C/-34.1,-29.3,-24.2,-19.9,-17.4,-13.8,-11.3,-8.9,-6.8,-5.0,
     1-4.0,-2.3,-1.1,-0.1,1.4,2.6,3.5,4.8,5.1,7.9,9.6,11.3,12.7,14.2,
     217.2,18.4,19.7/
      DATA D/0.0,0.07,0.14,0.21,0.28,0.42,0.56,0.71,0.85,1.0,1.14,1.29,
     11.44,1.57,1.71,1.85,1.98,2.12,2.25,2.39,2.53,2.66,2.79,2.92,3.04,
     23.17,3.28/
      DATA Q/0.0,0.0,0.0,0.0,0.0,0.1,0.1,0.4,1.3,2.2,2.9,3.6,5.1,6.3,7.5,
     19.1,10.9,12.5,15.0,17.5,19.6,24.0,27.9,31.7,36.9,43.0,48.7,52.3/
      DATA R/0.0,0.025,0.10,0.17,0.24,0.29,0.35,0.50,0.61,0.65,0.75,
     10.84,0.96,1.00,1.07,1.14/
      DATA S/0.0,0.25,1.00,1.25,2.00,3.00,3.60,5.25,7.00,8.10,11.50,14.5
     1,18.40,23.70,30.25,40.25/
      INDEX=INDEX+1
      DELAZ=AZ-GMAX
      DELEL=EL-GMAX                                                     SUB00100
      XD=99.9                                                           SUB00120
      DO 20 J=1,NIT                                                     SUB00130
      IF(DELAZ.GT.A(J))GO TO 20                                         SUB00140
      IF(DELAZ.EQ.A(J))GO TO 15
      IF(J.EQ.1)GO TO 14                                                SUB00160
      XA=A(J)-A(J-1)                                                    SUB00170
      XB=B(J)-B(J-1)                                                    SUB00180
      XC=DELAZ-A(J-1)                                                   SUB00190
      XD=((XC/XA)*XB)+B(J-1)                                            SUB00200
      GO TO 25
14    XD=0.0                                                            SUB00220
      GO TO 25                                                          SUB00230
15    XD=B(J)                                                           SUB00240
      GO TO 25
20    CONTINUE                                                          SUB00250
25    AB=DR*PHAZ                                                        SUB00260
      AD=COS(AB)
      IF(AD.LT.0.0)XD=-XD
      XH=99.9                                                           SUB00290
      DO 30 K=1,NNIT                                                    SUB00310
      IF(DELEL.GT.C(K))GO TO 30                                         SUB00320
      IF(DELEL.EQ.C(K))GO TO 32
      IF(K.EQ.1)GO TO 31                                                SUB00340
      XE=C(K)-C(K-1)                                                    SUB00350
      XF=D(K)-D(K-1)                                                    SUB00360
      XG=DELEL-C(K-1)
      XH=((XG/XE)*XF)+D(K-1)                                            SUB00380
```

```
      GO TO 35                                                          SUBC0390
   31 XH=0.0                                                            SUB00400
      GO TO 35                                                          SUB00410
   32 XH=D(J)                                                           SUB00420
      GO TO 35
   30 CONTINUE                                                          SUBC0430
   35 AF=DR*PHEL                                                        SUB00440
      AF=COS(AE)
      IF(AF.LT.0.0)XH=-XH
      CELEX=COS(ELEX)
      DELDG=SQRT((XH**2)+(((CELEX)*(XD))**2))                           SUBCC480
      XJ=99.9                                                           SUB00490
      DO 40 L=1,NITT                                                    SUBC0510
      IF(DELDG.GT.D(L))GO TO 40                                         SUBC0520
      IF(DELDG.EQ.D(L))GO TO 45                                         SUB00530
      XJ=((DELDG-D(L-1))/(D(L)-D(L-1)))*(Q(L)-Q(L-1))+Q(L-1)
      GO TO 41                                                          SUBC0550
   45 XJ=Q(L)                                                           SUB00560
      GO TO 41
   40 CONTINUE                                                          SUB00570
   41 XOHF=99.9
      NIX=16
      DO 50 M=1,NIX
      IF(DELDG.GT.R(M))GO TO 50
      IF(DELDG.EQ.R(M))GO TO 51
      XOHF=((DELDG-R(M-1))/(R(M)-R(M-1)))*(S(M)-S(M-1))+S(M-1)
      GO TO 52
   51 XOHF=S(M)
      GO TO 52
   50 CONTINUE
   52 AZE=AZE/DR
      ELEX=ELEX/DR                                                      SUB00590
      IHRX(INDEX)=IHR
      IMINX(INDEX)=IMIN
      TIM(INDEX)=AVGTM-TLIFT
      TEM(INDEX)=ZSEC
      GMAX(INDEX)=GMAX
      CRLC(INDEX)=GMAX+XJ                                               SUBC0620
      AVGRG=AVGRG+((IGAT*30.0)/1000.0)
      RF=AVGRG/XKMPT
      E=ELEX                                                            SUBC0640
      CALL REFC(E,RR,DEE,DRR)                                           SUBC0650
      BRANG(INDEX)=AVGRG-(DRR*XKMPT)
      RELEV(INDEX)=ELEX-DEE+XH
      AZCR(INDEX)=AZE+XD
      XF=XOHF
      IF(ICARD.EQ.0)GO TO 54
      WRITE(7,53)AVGTM,XK,XJ
   53 FORMAT(3F10.3)
   54 IF(INDEX.GT.1)GO TO 70
      WRITE(6,55)                                                       SUBC0700
   55 FORMAT(52X'DELTA  DELTA   DELTA   DELTA   DELTA')                 SUBC0710
      WRITE(6,60)
   60 FORMAT(8X'TIME      AZ     EL  V-LC(DB)  AZ(DB) EL(DB) AZ DEG EL DEGSUBC0730
     1  DEGREES VHF(DB) UHF(DB)     PHAZ       PHEL       CRLC')

   70 WRITE(6,75)IHR,IMIN,ZSEC,AZE,ELEX,GMAX,AZ,EL,XD,XH,DELDG,XJ,XF,PHA
     1Z,PHEL,CRLC(INDEX)
   75 FORMAT(2I3,F7.3,2F7.2,F6.2,3XF6.2,1XF6.2,2XF6.2,1XF6.2,3(2XF6.2),
     1 2F10.3,F10.3)
      RETURN
      END
```

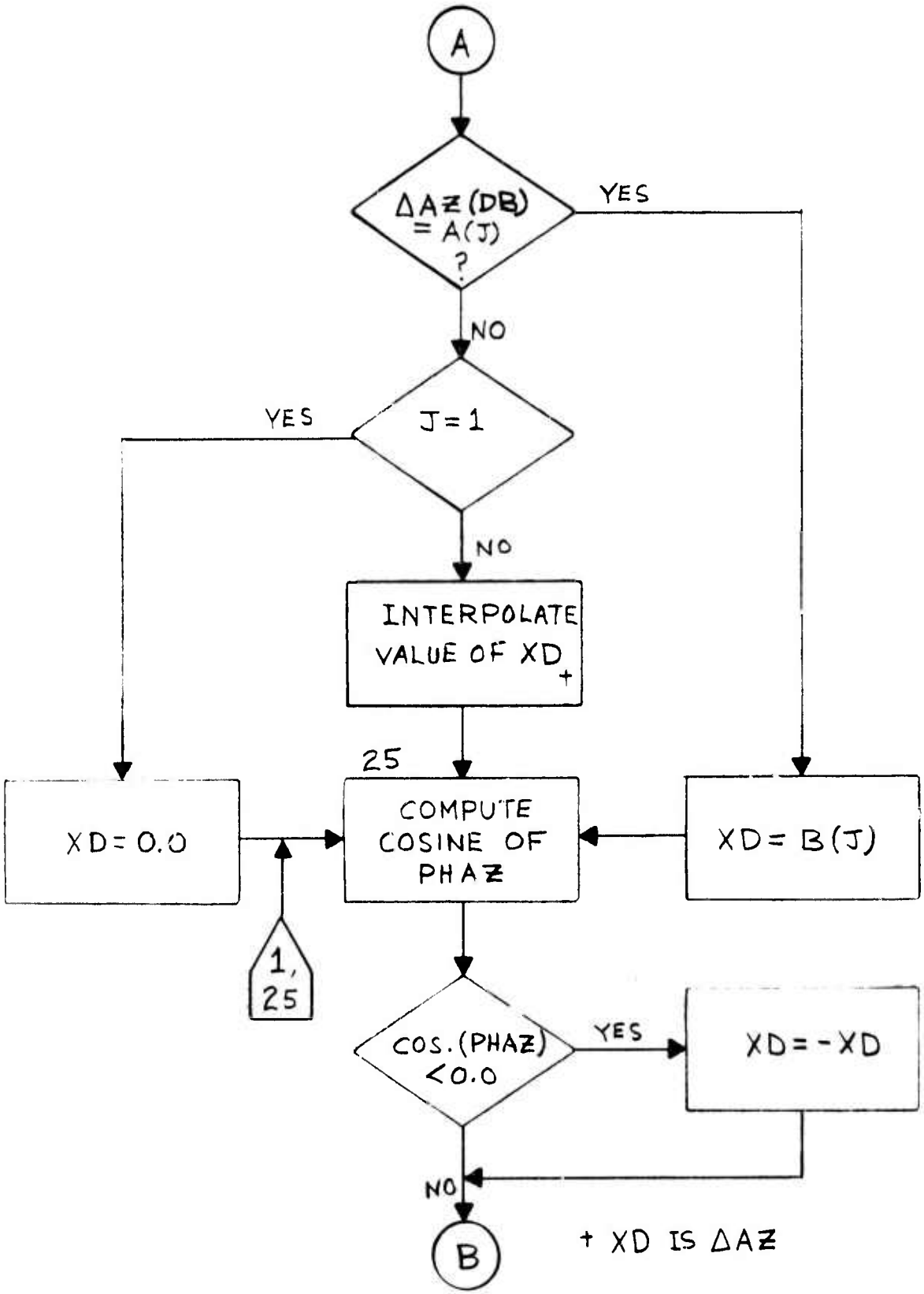
APPENDIX J

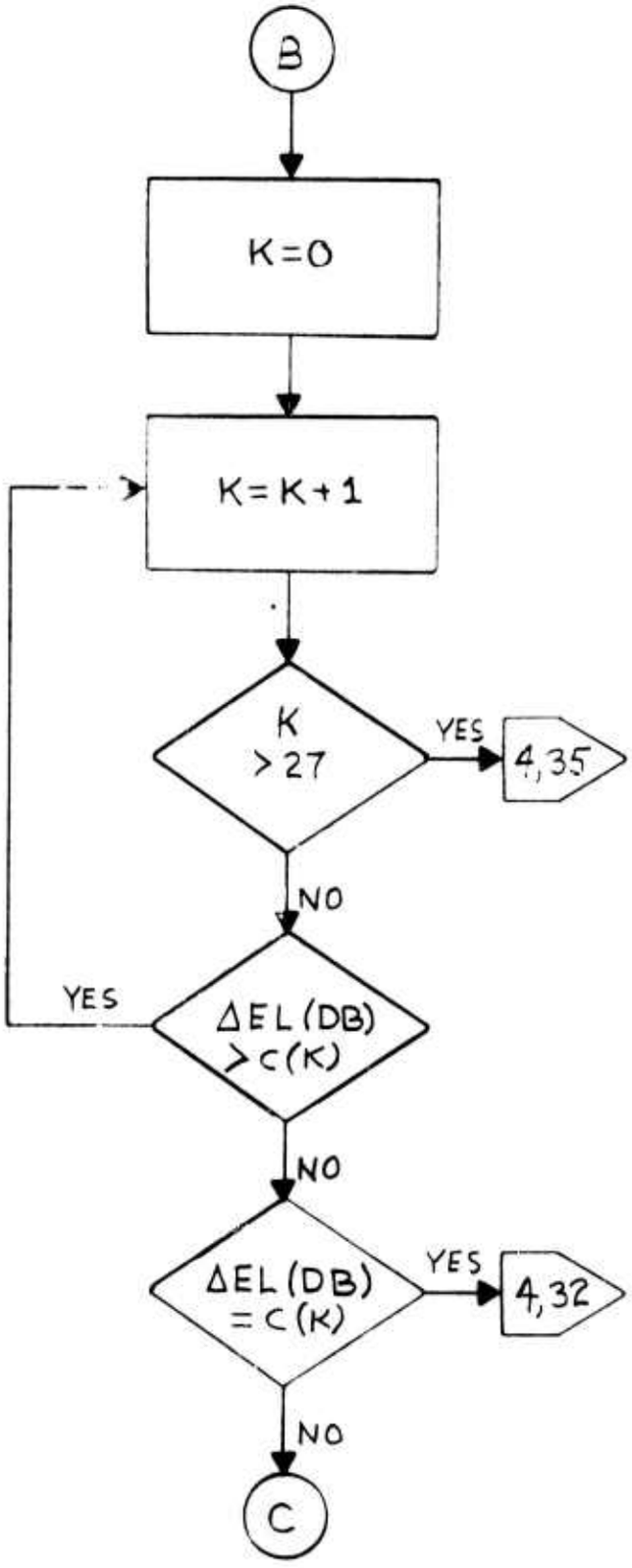
SUBROUTINE SUBOAK FLOW DIAGRAM

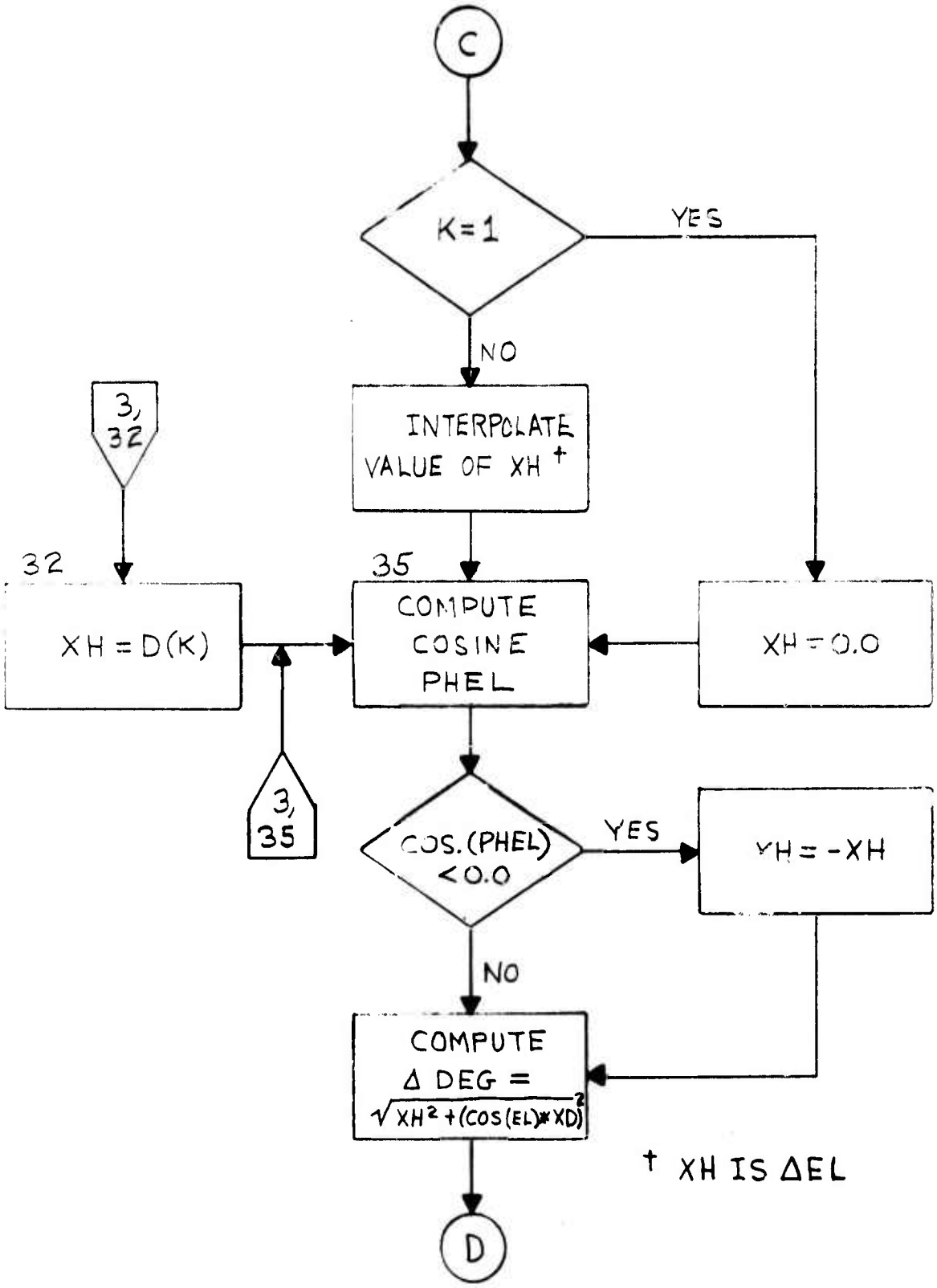


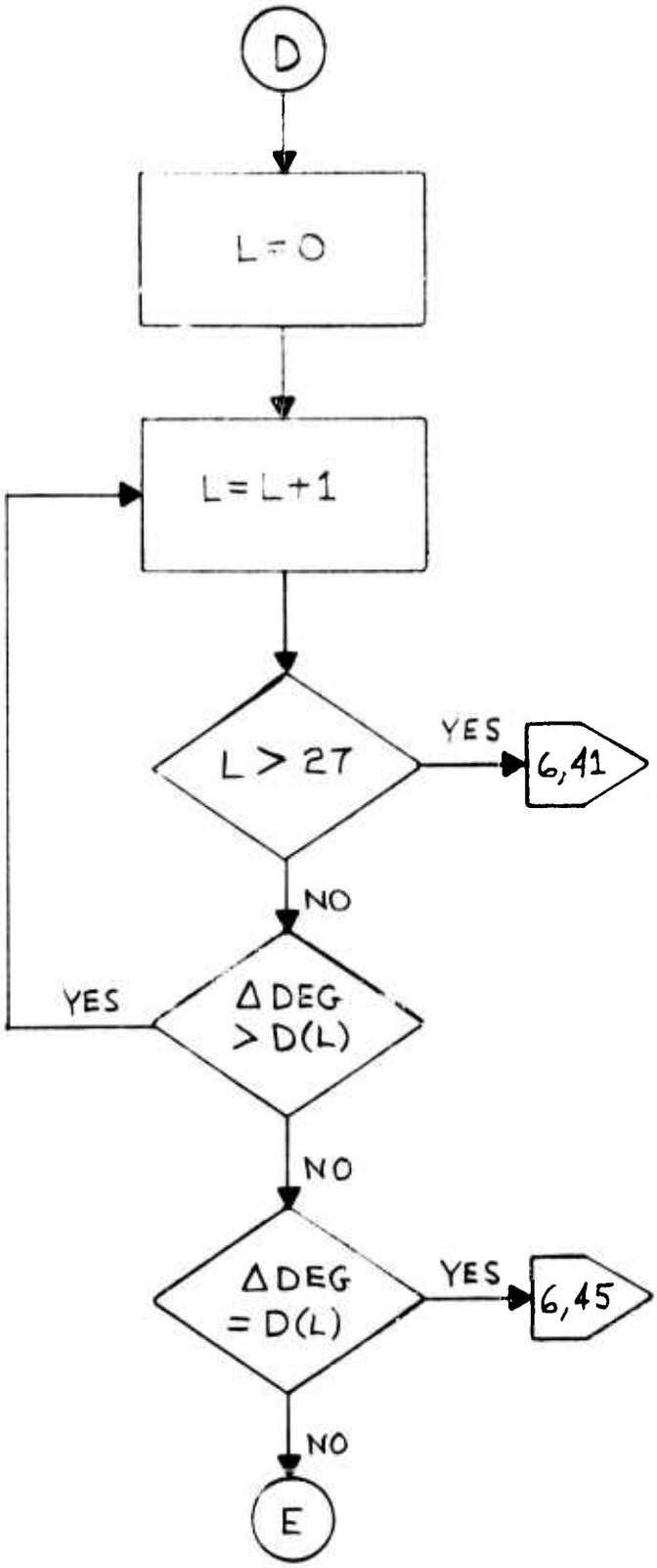
† THE TABLES CONTAIN:

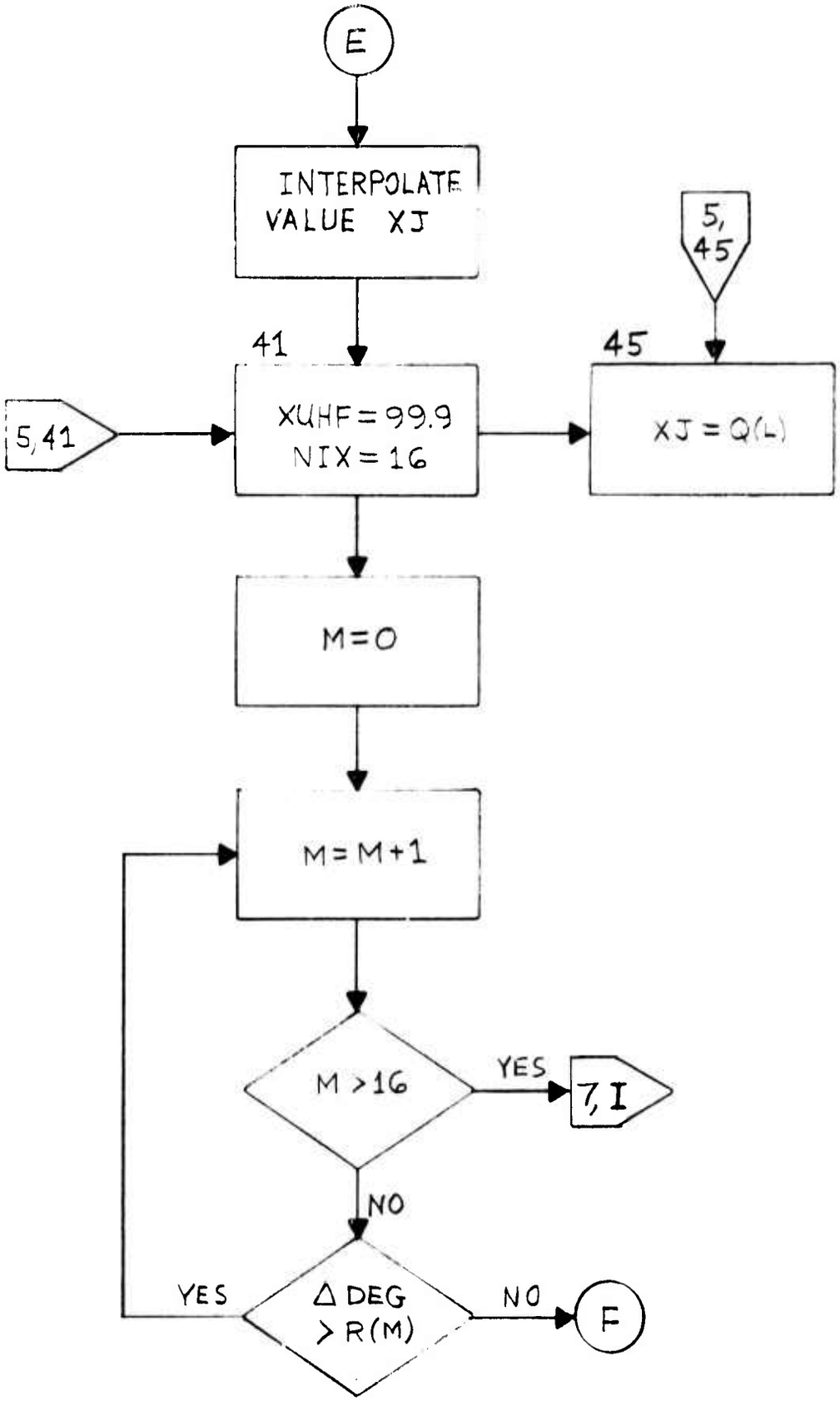
A	ΔAZ (DB)
B	ΔAZ
C	ΔEL (DB)
D	ΔEL & Δθ
Q	ΔVHF RCS (DB)
R	Δθ
S	ΔUHF RCS (DB)

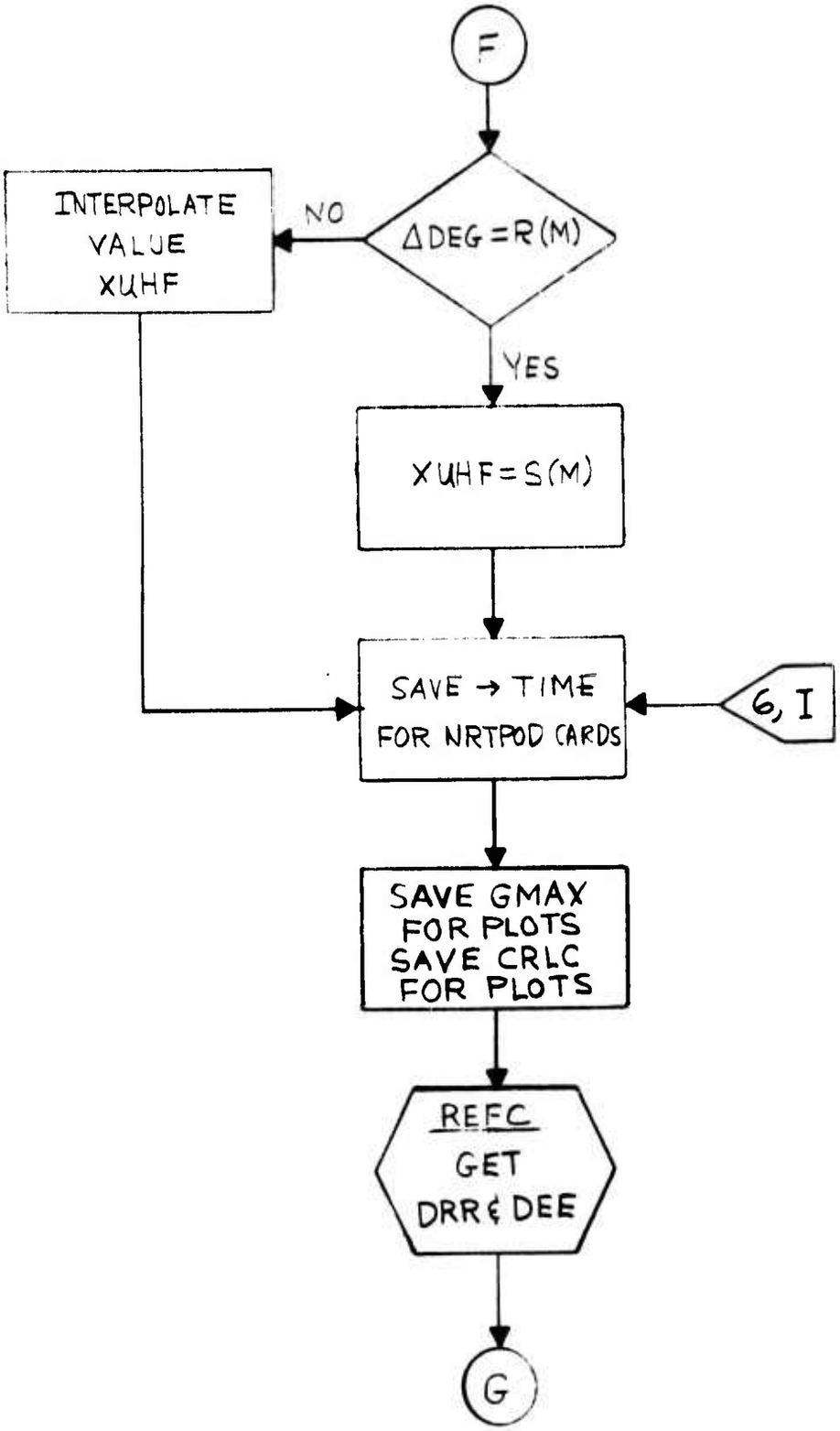

A
ΔAZ (DB) = A(J) ?
YES
NO
J = 1
YES
NO
INTERPOLATE VALUE OF XD+
25
XD = 0.0
COMPUTE COSINE OF PHAZ
XD = B(J)
1, 25
COS. (PHAZ) < 0.0
YES
XD = -XD
NO
B
+ XD IS ΔAZ

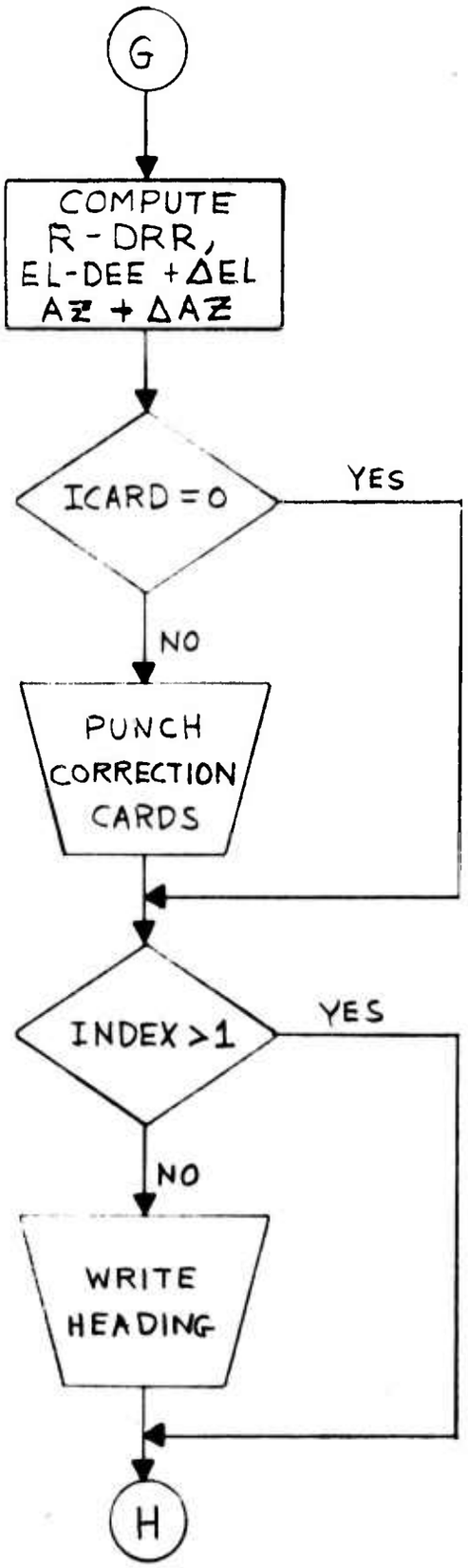

B
K=0
K=K+1
K
>27
YES
4,35
NO
YES
ΔEL(DB)
>C(K)
NO
ΔEL(DB)
=C(K)
YES
4,32
NO
C

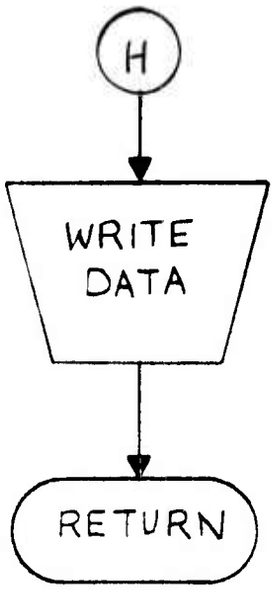

C
K=1
YES
NO
INTERPOLATE VALUE OF XH †
3, 32
32
XH = D(K)
35
COMPUTE COSINE PHEL
XH = 0.0
3, 35
COS.(PHEL) <0.0
YES
XH = -XH
NO
COMPUTE Δ DEG = √XH² + (COS(EL)*XD)²
D
† XH IS ΔEL


D
L=0
L=L+1
L > 27
YES
6,41
NO
YES
Δ DEG
> D(L)
NO
Δ DEG
= D(L)
YES
6,45
NO
E


E
INTERPOLATE VALUE XJ
5, 45
41
45
5, 41
XUHF = 99.9
NIX = 16
XJ = Q(L)
M = 0
M = M+1
M > 16
YES
7, I
NO
YES
Δ DEG > R(M)
NO
F


F
ΔDEG = R(M)
NO
INTERPOLATE VALUE XUHF
YES
XUHF = S(M)
SAVE → TIME FOR NRTPOD CARDS
6, I
SAVE GMAX FOR PLOTS
SAVE CRLC FOR PLOTS
REFC
GET DRR & DEE
G


G
COMPUTE
R - DRR,
EL - DEE + ΔEL
AZ + ΔAZ
ICARD = 0
YES
NO
PUNCH
CORRECTION
CARDS
INDEX > 1
YES
NO
WRITE
HEADING
H


H
WRITE
DATA
RETURN

APPENDIX K
SUBROUTINE REFC PROGRAM LISTING

VERSION 6/16/70

```
      SUBROUTINE REFC(E,R,DEE,DRR)                                          REFC0010
      DIMENSION DE(16,8),DR(16,8),ED(16),RD(8)                              REFC0020
      DATA DE/0.0   ,0.0   ,0.0   ,0.0   ,0.0   ,0.0   ,0.0   ,0.0   ,     REFC0030
     10.0   ,0.0   ,0.0   ,0.0   ,0.0   ,0.0   ,0.0   ,0.0   ,0.0313,     REFC0040
     20.0303,0.0292,0.0287,0.0282,0.0272,0.0262,0.0253,0.0243,0.0223,     REFC0050
     30.0214,0.0195,0.0171,0.0135,0.0075,0.0   ,0.0937,0.0848,0.0770,     REFC0060
     40.0732,0.0694,0.0627,0.0571,0.0522,0.0480,0.0412,0.0385,0.0337,     REFC0070
     50.0278,0.0205,0.0105,0.0   ,0.1850,0.1520,0.1250,0.1140,0.1050,     REFC0080
     60.0904,0.0795,0.0708,0.0636,0.0523,0.0478,0.0405,0.0323,0.0229,     REFC0090
     70.0114,0.0   ,0.5310,0.3070,0.2120,0.1830,0.1600,0.1280,0.1060,     REFC0100
     80.0899,0.0780,0.0612,0.0550,0.0455,0.0354,0.0246,0.0120,0.0   ,     REFC0110
     90.7550,0.3720,0.2400,0.2020,0.1750,0.1370,0.1120,0.0942,0.0811,     REFC0120
     A0.0631,0.0566,0.0466,0.0361,0.0250,0.0122,0.0   ,0.9120,0.4110,     REFC0130
     B0.2560,0.2140,0.1840,0.1420,0.1150,0.0967,0.0830,0.0643,0.0575,     REFC0140
     C0.0472,0.0365,0.0252,0.0122,0.0   ,0.9700,0.4200,0.2600,0.2200,     REFC0150
     D0.1900,0.1460,0.1170,0.0980,0.0840,0.0653,0.0584,0.0478,0.0369,     REFC0160
     E0.0254,0.0123,0.0 /                                                 REFC0170
      DATA DR/  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,     REFC0180
     1  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0, 22.6, 21.5, 20.4, 19.9,REFC0190
     2 19.4, 18.5, 17.6, 16.8, 16.1, 14.8, 14.2, 13.2, 12.0, 10.4,  8.6,REFC0200
     3  7.7, 67.3, 57.9, 50.2, 47.0, 44.1, 39.3, 35.4, 32.1, 29.3, 24.8,REFC0210
     4 22.9, 19.7, 16.3, 12.7,  9.4,  8.1,132.0, 98.5, 77.4, 69.7, 63.2,REFC0220
     5 52.9, 44.7, 38.4, 33.4, 26.4, 23.9, 20.1, 16.4, 12.7,  9.4,  8.1,REFC0230
     6340.0,167.0,103.0, 86.1, 73.4, 56.7, 46.2, 38.9, 33.6, 26.4, 24.0,REFC0240
     7 20.2, 16.4, 12.8,  9.5,  8.2,405.0,170.0,104.0, 86.3, 73.6, 56.8,REFC0250
     8 46.3, 38.9, 33.7, 26.5, 24.1, 20.3, 16.5, 12.8,  9.5,  8.2,421.0,REFC0260
     9171.0,104.0, 86.6, 73.9, 57.1, 46.4, 39.0, 33.8, 26.8, 24.3, 20.5,REFC0270
     A 16.6, 13.0,  9.8,  8.4,446.0,172.0,105.0, 87.4, 74.0, 58.0, 46.6,REFC0280
     B 39.2, 34.0, 27.0, 24.6, 20.7, 16.7, 13.0, 10.0,  8.4/              REFC0290
      DATA ED,RTDEG/0.01,2.0,4.0,5.0,6.0,8.0,10.0,12.0,14.0,18.,20.,      REFC0300
     124.,30.,40.,60.,90.,57.29578/                                       REFC0310
      DATA RD/0.01,10.,30.,60.,200.,400.,1000.,2000./                     REFC0320
      IF(R.LE.0.0)GO TO 300                                               REFC0330
      RG=R/6080.27                                                        REFC0340
      DO 100 IED=2,15                                                     REFC0350
      I=17-IED                                                            REFC0360
      IF(E.GE.ED(I))GO TO 120                                             REFC0370
  100 CONTINUE                                                            REFC0380
      I=1                                                                 REFC0390
  120 DO 200 JRD=2,8                                                      REFC0400
      J=10-JRD                                                            REFC0410
      IF(RG.GE.RD(J))GO TO 220                                            REFC0420
  200 CONTINUE                                                            REFC0430
      J=1                                                                 REFC0440
  220 IF(J.EQ.8)GO TO 340                                                 REFC0450
      ZR=ALOG(RG/RD(J))/ALOG(RD(J+1)/RD(J))                               REFC0460
      IF(E.LE.0.0)GO TO 320                                               REFC0470
      ZE=ALOG(E/ED(I))/ALOG(ED(I+1)/ED(I))                                REFC0480
      DE1=((DE(I+1,J)-DE(I,J))*(1.-ZR)+(DE(I,J+1)-DE(I,J))*ZR)*ZE         REFC0490
      DE2=((DE(I,J+1)-DE(I,J))*(1.-ZE)+(DE(I+1,J+1)-DE(I,J+1))*ZE)*ZR     REFC0500
      DEE=DE1+DE2+DE(I,J)                                                 REFC0510
      DR1=((DR(I+1,J)-DR(I,J))*(1.-ZR)+(DR(I,J+1)-DR(I,J))*ZR)*ZE         REFC0520
      DR2=((DR(I,J+1)-DR(I,J))*(1.-ZE)+(DR(I+1,J+1)-DR(I,J+1))*ZE)*ZR     REFC0530
      DRR=(DR1+DR2+DR(I,J))                                               REFC0540
      GO TO 400                                                           REFC0550
  300 DEE=0.0                                                             REFC0560
      DRR=0.0                                                             REFC0570
      GO TO 400                                                           REFC0580
  320 DEE=DE(I,J)+(DE(I,J+1)-DE(I,J))*ZR                                  REFC0590
      DRR=DR(I,J)+(DR(I,J+1)-DR(I,J))*ZR                                  REFC0600
      GO TO 400                                                           REFC0610
  340 DELT=(E-ED(I))/(ED(I+1)-ED(I))                                      REFC0620
      DEE=DELT*(DE(I+1,J)-DE(I,J))+DE(I,J)                                REFC0630
      DRR=DELT*(DR(I+1,J)-DR(I,J))+DR(I,J)                                REFC0640
  400 RETURN                                                              REFC0650
      END                                                                 REFC0660
```

APPENDIX L

SUBROUTINE ALREAD PROGRAM LISTING

```
      SUBROUTINE ALREAD(TSTART,TSTOP,TLIFT,INTARG,INPAT,NOPHA,NPTS,NFP,
     1NEWPAS,NRG,ISTGAT)
      DIMENSION ALT(5),AMT(3,5),AZ(6),CALADD(4),CALCON(19),EL(6),IHD(13)
     1,IMT(5),INGATE(5,3),IPRICE(5),ISENS(6),ISLIDE(5,3),ISPAC(5,3),
     2ITARDT(5,3,3),ITARG(5),IWAVE(6),LOC(5,3,4),MODE(5,3),NC(4),NMX(2),
     3NMODES(5,3),NPTEST(2),NSAMP(5,3),NSAMPT(5),POWER(6),RAD(6,5),
     4RANGE(5),VEL(5),IFPAR(22)
      COMMON/TREAD/LN,IFLG,IBTRHD,FMTRHD,FMTRMA,FMTRTG,FMTRFI,FMTRSP,
     1FMXSEC,FMARSG,FMRR11,FMGLOT,FMCHAF,FMBSMC,FMASLP,FMAMP(6),FMPHA(6)
     2,NAME(25),NI(24),IX(24),TAMP(128,6),TPH(128,6),ITEM(2000)
      COMMON/TIMCOM/IHMS,IFS
      COMMON/RDCOMT/TIMES(300),XSPHA(4,30,300),RANGKM(300),ALSAV(300),
     1AZI(300),ELE(300),IRGA(30),IPOL(4),NPOL
      EQUIVALENCE (IHD(1),IDREC),(IHD(2),LREC),(IHD(3),ITGTM1),
     1(IHD(4),ITGTM2),(IHD(5),IFPG),(IHD(6),IPRI),(IHD(7),NELRD),
     2(IHD(8),MACYBA),(IHD(9),NTARG),(IHD(10),NTDBA),(IHD(11),NMINOR),
     3(IHD(12),MICYBA),(IHD(13),LMICY),(IFPAR(1),FMTRHD),
     4(RAD(1,1),POWER(1)),(RAD(1,2),AZ(1)),(RAD(1,3),EL(1)),
     5(ITARDT(1,1,1),MODE(1,1)),(ITARDT(1,1,2),ISPAC(1,1)),
     6(ITARDT(1,1,3),NSAMP(1,1)),(IMT(1),IGCHG),(IMT(2),IGAIN),
     7(IMT(3),NPLS),(IMT(4),ISLBA),(IMT(5),NBP),(AMT(1,1),RANGE(1)),
     8(AMT(1,2),VEL(1)),(AMT(1,3),ALT(1))
      DATA PCON,RKM,VKM/1.0E6,1.873703E-3,4.4672E-4/
      DATA NEX,NPTEST/0,1,300,150/
      DOUBLE PRECISION DRANG,FINTIM,FPG,GMTIME,PRF,TIME(6),TIMES,TIMOLD,
     1TLIFT,TSTART,TSTOP,2TDIF(6)
      INTEGER*2 ITEM
      IF(NEWPAS.GT.2)GO TO 2000
      IF(NRG.GT.30)NRG=30
      NPOL=0
      DO 10 I=1,4
      IF(IPOL(I).NE.0)NPOL=NPOL+1
10    CONTINUE
      GMTIME=TSTART-1.0
      NPTS=0
      TINC=0.0
      NTERR=0
      CALL THEAD(NEWPAS,&9960)
      DO 60 I=1,6
      IF(IFPAR(I).GT.0)GO TO 60
      WRITE(6,40)NAME(I)
40    FORMAT(' FORMAT TABLE ',A4,' WAS NOT FOUND - RUN ABORTED.')
      NEWPAS=55
      RETURN
60    CONTINUE
80    DO 100 I=2,19
      CALCON(I)=GET(FMXSEC,IBTRHD,I)
100   CONTINUE
      TLIFT=0.0
      ITL=IGET(FMGLOT,IBTRHD,1)
      IF(ITL.NE.2)GO TO 140
      IHMS=IGET(FMGLOT,IBTRHD,2)
      IFS=IGET(FMGLOT,IBTRHD,3)
      CALL GMTUPK(TLIFT)
      GO TO 140
```

```
120   CALL BREAD(1)
140   IHD(1)=IGET(FMTRHD,IBTRHD,1)
      IF((IDREC.LT.127).AND.(IFIG.NE.3))GO TO 180
      WRITE(6,160)GMTIME
160   FORMAT(' END OF FILE ENCCUNTERED ON INPUT TAPE AFTER TIME ',F12.4)
      NEWPAS=44
      RETURN
180   IF(IDREC.NE.2)GO TO 120
      TIMOLD=GMTIME
      DO 200 I=2,13
      IHD(I)=IGET(FMTRHD,IBTRHD,I)
200   CONTINUE
      IHMS=IGET(FMTRHD,IBTRHD,14)
      IFS=IGET(FMTRHD,IBTRHD,15)
      CALL GMTUPK(GMTIME)
      IF(GMTIME.LT.(TSTART-0.20))GO TO 120
      IF((GMTIME.GT.TIMOLD).AND.(MACYBA.EQ.0).AND.(NTDBA.EQ.0))GO TO 120
      NCONT=0
      IF(((DABS(TIMOLD-GMTIME)).GT.0.01).AND.(NTDBA.NE.0))GO TO 280
      IF(NTERR.NE.0)GO TO 120
      NCONT=1
      GO TO 1200
280   FINTIM=GMTIME
      TIMOLD=GMTIME
      NFP=IFPG+1
      DO 300 I=1,5
      NSAMPT(I)=0
300   CONTINUE
      PRF=FCON/FLOAT(IPRI)
      TINK=1./PRF
      IF(NBLRD.EQ.0)GO TO 600
      NBLRDC=NBLRD
      DO 500 I=1,NBLRD
      IBTRMA=IBTRHD+MACYBA+24*(I-1)
      DO 400 K=1,3
      RAD(I,K)=GET(FMTRMA,IETRMA,K)
400   CONTINUE
      IHMS=IGET(FMTRMA,IBTRMA,4)
      IFS=IGET(FMTRMA,IETRMA,5)
      CALL GMTUPK(TIME(I))
      ISENS(I)=IGET(FMTRMA,IBTRMA,6)
      IWAVE(I)=IGET(FMTRMA,IBTRMA,7)
      IF(IWAVE(I).EQ.4)IWAVE(I)=3
      IF(IWAVE(I).GT.3)IWAVE(I)=0
500   CONTINUE
      NCHANZ=IFPG
      IF(IFPG.EQ.1)NCHANZ=NCHANZ+3
      DO 560 IP=1,NPCL
      NC(IP)=NCHANZ+IPCL(IP)
      DO 520 I=1,NOPHA
      INDEX=(I-1)*6+NC(IP)+12
      IF(IFPAR(INDEX).GT.0)GO TO 520
      WRITE(6,40)NAME(INDEX)
      NEWPAS=66
      RETURN
```

```
520   CONTINUE
      KCNGET=3*(NC(IP)-1)+IWAVE(1)+1
      CALADD(IP)=CALCON(KCNGET)-POWER(1)-FLOAT(10*ISENS(1))
560   CONTINUE
600   IF(NTARG.EQ.0)GO TO 120
      ITPIK=0
      NBPTST=0
      DO 900 I=1,NTARG
      IBTRTG=IBTRHD+NTDBA+12*(I-1)
      ITARG(I)=IGET(FMTRTG,IBTRTG,1)
      IPRICR(I)=IGET(FMTRTG,IBTRTG,2)
      NK=1
      DO 800 K=1,3
      IK=NK+1
      NK=IK+2
      DO 700 L=1,3
      ITARDT(I,K,L)=IGET(FMTRTG,IBTRTG,I+IK)
700   CONTINUE
      IF(ISPAC(I,K).EQ.254)ISPAC(I,K)=-1
      ISPAC(I,K)=2**(ISPAC(I,K)+NEX(IFPG+1))
      IF(ISPAC(I,K).EQ.0)ISPAC(I,K)=1
      NSAMPT(I)=NSAMPT(I)+NSAMP(I,K)
      NMODES(I,K)=0
      DO 780 L=1,4
      LOC(I,K,L)=0
      MODUM=MOD(MODE(I,K),2)
      IF(MODUM.EQ.0)GO TO 760
      NMODES(I,K)=NMODES(I,K)+1
      LOC(I,K,L)=NMODES(I,K)
760   MODE(I,K)=MODE(I,K)/2
780   CONTINUE
      ISLIDE(I,K)=2*NSAMP(I,K)*NMODES(I,K)
      NBPTST=NBPTST+ISLIDE(I,K)
800   CONTINUE
      DO 820 J=1,3
      INGATE(I,J)=IGET(FMTRTG,IBTRTG,J+11)
820   CONTINUE
      IF(ITARG(I).EQ.INTARG)ITPIK=I
900   CONTINUE
      NBPTST=NBPTST-MOD(NBPTST,6)
      NTERR=0
      IF(ITPIK.GT.0)GO TO 960
      WRITE(6,920)GMTIME,INTARG,(ITARG(I),I=1,NTARG)
920   FORMAT('  AT TIME = ',F12.4,' TARGET ',I2,' IS NOT ON THE TAPE, TA
     1RGETS AVAILABLE ARE ',5I3)
      IBTRMI=IBTRHD+MICYEA
      NBP=IGET(FMTRMI,IBTRMI,5)
      IF(NBP.EQ.NBPTST)GO TO 10000
      NTERR=1
      GO TO 120
960   IF(NMODES(ITPIK,INPAT).GE.NPOL)GO TO 1020
      WRITE(6,1000)
1000  FORMAT(' EITHER POLARIZATION OR PATTERN CHOSEN IS NOT AVAILABLE')
      GO TO 10000
1020  IF(ISTGAT.GT.NSAMP(ITPIK,INPAT))ISTGAT=1
```

```
      IF(ISTGAT.LT.1)ISTGAT=1
      NSTEST=NSAMP(ITPIK,INPAT)-ISTGAT+1
      NSTGAT=ISTGAT
      IPAT=INPAT
      I=1
1040  IRGA(I)=INGATE(ITPIK,IPAT)+(NSTGAT-1)*ISPAC(ITPIK,IPAT)
      NSTGAT=1
1060  IF(I.GE.NRG)GO TO 1100
      IF((I+1).GT.NSTEST)GO TO 1080
      I=I+1
      IRGA(I)=IRGA(I-1)+ISPAC(ITPIK,IPAT)
      GO TO 1060
1080  IPAT=IPAT+1
      IF(IPAT.GT.3)GO TO 1100
      IF(NSAMP(ITPIK,IPAT).EQ.0)GO TO 1080
      NSTEST=NSTEST+NSAMP(ITPIK,IPAT)
      IF(NMODES(ITPIK,IPAT).LT.NPOL)GO TO 1100
      I=I+1
      IF(I.LE.NRG)GO TO 1040
      I=I-1
1100  IPAT=INPAT
      NRG=I
1200  IBTRMI=IBTRHD+MICYBA
      DO 2200 MIN=1,NMINOR
      IF(MIN.LE.1)GO TO 1220
      IBTRMI=IBTRMI+LMICY+NPLS*NBP
1220  DO 1240 I=1,5
      AMT(I)=IGET(FMTRMI,IBTRMI,I)
1240  CONTINUE
      IF(NBP.EQ.NBPTST)GO TO 1280
      WRITE(6,1260)GMTIME,MIN,NBP,NBPTST
1260  FORMAT(' AT TIME = ',F10.4,', MINOR CYCLE ',I1,' SOMETHING IS SCRE
     1WED UP NBP = ',I5,' IT SHOULD BE ',I4,' - SHOOT HARTOGENSIS')
      GO TO 1330
1280  IF(NILS.GE.1)GO TO 1340
      WRITE(6,1320)NPLS,MIN,NMINOR
1320  FORMAT(' NPLS = ',I2,3X,'MINOR CYCLE ',I1,3X,I1,' MINOR CYCLES')
1330  GMTIME=GMTIME+TINC
      GO TO 120
1340  TINC=FLOAT(NPLS)/PRF
      DO 1500 L=1,NTARG
      IBTRMT=IBTRMI+(L-1)*12
      DO 1400 I=1,3
      AMT(I,I)=GET(FMTRMI,IBTRMT,I+5)
1400  CONTINUE
      VEL(L)=VEL(L)*VKM
1500  CONTINUE
      IBTRMT=IBTRMI+(ITPIK-1)*12
      TSDREI=IGET(FMTRMI,IBTRMT,9)
      NSDUM=NSAMPI(ITPIK)
      IF(NRG.LT.NSDUM)NSDUM=NRG
      DRANG=RANGE(ITPIK)
      IF(DRANG.GT.0.0)GO TO 1560
      WRITE(6,1540)GMTIME,DRANG
1540  FORMAT('  AT TIME = ',F15.4,' THE RANGE = ',1PE20.6)
```

```
      DRANG=1.0
1560  DRANG=40.*DLOG10(DRANG)
      CALIB=DRANG+FLOAT(IGAIN)
      DO 2000 I=1,NPLS
      IF((MIN.GT.1).OR.(I.GT.1))FINTIM=FINTIM+TINK
      IF(ISTART.GT.FINTIM)GO TO 2000
      NPTS=NPTS+1
      IF(NPTS.GT.1)GO TO 1600
1580  TIMES(NPTS)=FINTIM
      GO TO 1620
1600  IF((I.EQ.1).AND.(MIN.EQ.1).AND.(NCONT.EQ.0))GO TO 1580
      TIMES(NPTS)=TIMES(NPTS-1)+TINK
1620  RANGKM(NPTS)=RANGE(ITPIK)*RKM+VEL(ITPIK)*(TIMES(NPTS)-FINTIM)
      ALSAV(NPTS)=ALT(ITPIK)
      IF(NBLRD.NE.0)GO TO 1630
      INTERP=NBLRD
      GO TO 1650
1630  ZTDIF(1)=DABS(TIMES(NPTS)-TIME(1))
      INTERP=1
      DO 1640 NAE=2,NBLRD
      ZTDIF(NAE)=DABS(TIMES(NPTS)-TIME(NAE))
      IF(ZTDIF(NAE).LE.ZTDIF(NAE-1))INTERP=NAE
1640  CONTINUE
1650  AZI(NPTS)=AZ(INTERP)
      ELE(NPTS)=EL(INTERP)
      ISAMPT=IBTRHD+ISDBA+ISDREL+(I-1)*NBP
      NSTEST=NSAMP(ITPIK,INPAT)-ISTGAT+1
      IAD=0
      IF(INPAT.EQ.1)GO TO 1680
      JST=INPAT-1
      DO 1660 J=1,JST
      IAD=IAD+ISLIDE(ITPIK,J)
1660  CONTINUE
1680  L=ISTGAT-1
      DO 1800 K=1,NRG
      L=L+1
      IF(K.LE.NSTEST)GO TO 1700
      IAD=IAD+ISLIDE(ITPIK,IPAT)
      IPAT=IPAT+1
      NSTEST=NSTEST+NSAMP(ITPIK,IPAT)
      L=1
1700  DO 1780 IP=1,NPOL
      IPIK=ISAMPT+IAD+2*(LOC(ITPIK,IPAT,IPOL(IP))-1+NMODES(ITPIK,IPAT)
     1*(L-1))
      IAMP=IGET(FMTRSP,IPIK,1)
      IF((IAMP.LT.1).OR.(IAMP.GT.128))IAMP=1
      XSPHA(IP,K,NPTS)=TAMP(IAMP,NC(IP))+CALIB+CALADD(IP)
1740  IF(NCPHA.EQ.1)GO TO 1780
      IPHA=IGET(FMTRSP,IPIK,2)
      IF((IPHA.GE.0).AND.(IPHA.LE.127))GO TO 1760
      XSPHA(IP,K,NPTS+150)=0.0
      GO TO 1780
1760  XSPHA(IP,K,NPTS+150)=TPH(IPHA+1,NC(IP))
1780  CONTINUE
1800  CONTINUE
```

```
      IPAT=INPAT
      IF(ISTOP.LT.TIMES(NPTS))GO TO 10000
      IF(NPTS.LT.NPTEST(NCPHA))GO TO 2000
      NEWPAS=99
      RETURN
2000  CONTINUE
2200  CONTINUE
      GO TO 120
9960  WRITE(6,9980)
9980  FORMAT('  THEAD HAS DEFAULTED - RUN HAS BEEN ABORTED.')
10000 NEWPAS=0
      RETURN
      END
```